

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representation of
The original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

THIS PAGE BLANK (USPTO)



(19) Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 735 749 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.10.1996 Bulletin 1996/40

(51) Int Cl. 6: H04N 5/445

(21) Application number: 96301902.1

(22) Date of filing: 20.03.1996

(84) Designated Contracting States:
DE FR GB

(30) Priority: 28.03.1995 US 412393

(71) Applicant: AT&T IPM Corp.
Coral Gables, Florida 33134 (US)

(72) Inventors:
• Eick, Stephen Gregory
Naperville, Illinois 60565 (US)

• Walpole, Rebecca Anne
Corvallis, Oregon 97330 (US)
• Mataga, Peter Andrew
Naperville, Illinois 60563 (US)

(74) Representative:
Buckley, Christopher Simon Thirsk et al
Lucent Technologies,
5 Mornington Road
Woodford Green, Essex IG8 0TU (GB)

(54) Method and apparatus for finding and selecting a desired data item from a large schedule of data items using a TV set and a controller similar to a TV-remote-control

(57) An apparatus and method for presenting a viewer with an overall representation of the present number of entertainment programs available for selection given one week of program schedule data for 300 or more channels and one or more filtering criteria to limit the number of items represented in the overall representation. Sequentially applied filters will filter the group of program schedule data items that has at least 100,000 half hour time slots offered by 300 channels each week into a smaller subgroup where individual

consideration of each item of the subgroup can be made in a reasonable time. A set top box drives the display of overall representations or results of filtering criteria on a commercial TV set. Once a reasonable sized subgroup is obtained, other displays provide specific information of the program offerings of the subgroup. Selection of the filtering criteria and selection from within a subgroup is interactively made by a viewer through the use of a controller that looks and operates very much like a TV remote control. This makes the interaction familiar, easy and predictable.

EP 0 735 749 A2

Printed by Jouve, 75001 PARIS (FR)

Description

Technical Field

5 The invention concerns a method and apparatus for subjecting a large schedule of data items having multiple attributes to consecutive selection criteria in order to reduce the number of individual programs to a manageable group which can be visually searched for a desired data item having a selected subset of the attributes, and more particularly to an apparatus and method which use an interactive control having directional buttons and a select button that are used in conjunction with an interactive display viewed on a normal television set to select the desired data item.

10

Description of the Prior Art

15 Presently there are known methods for reducing a large quantity of data into a manageable set of data which can be visually searched for a desired item by a decision maker. One example of such a large quantity of data is a directory of a fixed drive of a computer system. Methods implemented through interactive graphical user interfaces for personal computers and workstations display and reduce disk drive directories to root directory displays which typically show root level files and one or more branch subdirectories for the user's selection. Upon selection of a subdirectory, usually by a mouse, the display typically shifts showing files of the selected subdirectory and sub-subdirectories for further selection. The subdirectory display is often too big to fit on the screen, so interactive scroll bars are typically provided so the display may be controlled by a mouse. Using the mouse and the scroll bars, a user may work down the directory tree structure until the desired file is found. Such graphical user interfaces are common for computers and monitors where visual definition is typically at least 640 x 450 pixels for each display. Such techniques might be used in homes to access databases of useful information, such as airline schedules, television programming schedules and movie-on-demand catalogues. Unfortunately, each home does not have a computer or work station with 640x480 pixel definition which could take advantage of such existing databases. Further, the NTSC television set which almost every home has in its living room has relatively low viewing definition compared to 640 x 450 pixels or more per screen definition of computer monitors. Moreover, the typical home television set is not connected to a mouse, which is not an appropriate pointing device for the living room, rather most television sets have controls on control panels and/or on a remote controls. If just a fraction of these home television sets were used to find and select airline ticket reservations, programs to watch on 300 hundred or more channel cable television services, or pay-per-view movies from a vast collection, the profitability of the service providers and the satisfaction of the users would both be improved. The 300 plus channels mentioned, may use any type of transmission scheme that will deliver information via a cable or wireless path and includes but is not limited to time division multiplexed channels, frequency division multiplexed channels and packet data multiplexed channels.

20

30 One known approach for the TV programming schedule is to display the presently showing programs along with the next subsequent programs for the next hour or so, on what is referred to as a preview channel. Because this is more information than can be legibly displayed on one television screen at once, the preview channel display often scrolls through all the channel offering for the present time and the near future. For a sixty channel system, one complete scrolling takes about three minutes. At such a rate, a one hundred channel cable service would take five minutes and the future three hundred plus channel cable services would take 15 minutes. Needless to say, three minutes is a long time, but acceptable because breaks between programs are about that long. Five and fifteen minutes time periods though represent a substantial portion of a 30 minute program and are simply too long to expect a TV viewer to wait. The alternatives of speeding up the scrolling rate or using smaller size letters for descriptions are not practical either because either of these actions reduces the ability of the viewer to read and understand the schedule. Thus, there is a need in the art for a method and apparatus that allows a viewer to quickly find and select a desired data item from a large schedule, in this case a TV program for viewing from a TV programming schedule for 300 plus channels over the ensuing hours or even days. There is a similar need for a method and apparatus, very similar to the TV program selector, for finding and selecting a movie to order from movies-on-demand, or an airline flight(s) for a trip. It would further be desirable to use a method similar to the TV program selector to find and select a file in storage assets accessible by the apparatus to be executed, updated or deleted as part of file maintenance.

35

40 It is an object of the present invention to provide a view of a large schedule of data items and interactive selections of subgroups of the large schedule of data items in order to arrive at a screen display with sufficiently small number of items and sufficiently legible descriptions of each item to provide a viewer with an opportunity to make a reasoned selection therefrom.

45

50 It is another object of this invention to provide a method for interactively selecting a data item from a large schedule of data items by means of sequentially applying different filtering criteria using an interactive control having an operation appropriate for use with a television set.

Summary of the Invention

5 In another aspect of the invention, the aforementioned objects may be achieved by providing a method for a home television viewer to interactively select a data item from a large schedule of data item having multiple attributes. The method includes a step of receiving the large schedule of data items. The received schedule of data items is stored locally in a database format in order to expedite later filtering and retrieval. Next, the schedule of data items is filtered into a subgroup of the schedule of data items according to attributes selected by to interactive viewer inputs. The resulting subgroup of the schedule data items is displayed for the viewer's inspection. The user then interactively selects a data item from the subgroup of data items viewed on a television screen.

10 Briefly stated, in accordance with one aspect of the invention, the aforementioned objects are achieved by providing an apparatus for selecting an item from a large group in a system having display means and interactive movable pointing means for specifying a location in the display means and making a selection at a specified location. This apparatus includes a filtration means including subgroup specifiers in the display means and is responsive to selection of a subgroup specifier by the pointing means for filtering the list to produce the subgroup specified by the selected subgroup specifier; means for displaying representations of group items belonging to at least a portion of the subgroup in the display means; and group item selection means for selecting a group item by selecting the representation thereof in the display in response to the pointing means.

15 In yet another aspect of the invention, the aforementioned objects may be achieved by providing a method for a viewer to interactively select a program. The method includes a step of receiving program schedule data for at least 300 individual channels for a time period of at least a week. The received program schedule data is stored locally in a database format in order to expedite later sorting and retrieval. Next, the program schedule data is filtered into a subgroup of the program schedule data in response to interactive viewer inputs. The subgroup of the program schedule data is displayed for the viewer's inspection. The user then interactively selects a program from the subgroup of program schedule data for viewing on a TV screen, or alternatively for recording by an appropriate program recording device.

20 Brief Description of the Drawing

25 FIG. 1 is a pictorial of a television set connected through a set top box to a cable carrying the program to be selected and a controller for selecting that program.

30 FIG. 2 is a simplified block diagram of the set top box.

FIG. 3 is a pictorial of a controller as shown in FIG. 1.

FIG. 4 is a pictorial of a top most selection interactive display.

FIG. 5 is a pictorial of a second level selection interactive display.

FIG. 6 is a pictorial of a third level selection interactive display.

FIG. 7 is a pictorial of a first level selection query display.

FIG. 8 is a pictorial of a second level selection query display.

FIG. 9 is a pictorial of another third level selection query display.

FIG. 10 is a pictorial of a display showing a subgroup of programs meeting the Sports, All and On Now sorting criteria.

40 FIG. 11 is a pictorial of a display showing the subgroup of programs meeting the Sports, All and On Now sorting criteria along with a window having a preview of the highlighted program.

FIG. 12 is a pictorial of a display showing a second level selection interaction display, similar to FIG. 5.

FIG. 13 is a pictorial of a two-dimensional interactive grid display with very many program data items shown in reduced representations.

45 FIG. 14 is a pictorial of a third level selection query display, similar to FIG. 9.

FIG. 15 is a pictorial of a two-dimensional interactive grid display filtered down to a manageable number of data items.

FIG. 16 is a pictorial of a first alphanumeric interactive display.

FIG. 17 is the same display as FIG. 14 except that the highlighted interactive area is at a different location.

50 FIG. 18 is a pictorial of a second alphanumeric interactive display.

FIG. 19 is the same display as FIG. 16 except the highlighted interactive area is at a different location.

FIG. 20 is a pictorial of a third alphanumeric interactive display.

FIG. 21 is the same as FIG. 18 except that the highlighted interactive area is at a different location.

FIG. 22 is a pictorial of a fourth alphanumeric interactive display.

55 FIG. 23 is a pictorial of a two dimensional interactive display with logical third dimensional stacks for row and column intersections having multiple entries therein.

buttons 52-56 and double arrow buttons 60, 62 of controller 20. The file card menu 402 is surrounded by a frame 420, the top of which indicates the designation of the active area currently highlighted. Once an active area has been highlighted, a selection is made by actuating the **select (✓)** button 64 in FIG. 4, the **TV** button 410 is shown to be active; by actuating the **select (✓)** button 64, the next display 500 shown in FIG. 5 appears. This appearance is a logical overlaying of the display 500 over the display 400. Although display 400 is not visible while any logically overlaying display is appearing on the screen of the TV 10, display 400 will become visible again if all of the logically overlaying displays are canceled, i.e. by actuating the **cancel (X)** button 66. Thus, until a program is selected for real time viewing, it is possible for the viewer to work his or her way back to the display 400 by actuating the **cancel (X)** button the appropriate number of times.

FIG. 5 shows a second level display 500 which is depicted as a file card menu 502 labeled "TV", which appears to overlay and occlude all of file card menu 402 except for the label "Begin". The label **TV** indicates that the items that can be accessed are TV shows, such as dramatic series, situation comedies, serials, regular variety shows, game shows, sports, and so forth. Since movies and shopping were topics of other interactive buttons, these types of programs may be filtered out in whole or in part. File card menu 502 has interactive buttons labeled **On Now** 504, **Weekdays** 506, **Coming Up** 508, **Weekend** 510, and **Search** 512. As with the file card menu 402, file card menu 502 has an active area that can be moved by the viewer by operation of the arrow buttons 52-56 and double arrow buttons 60, 62 of controller 20 (shown in FIG. 3). Each of the interactive button represents another filtering that will be performed if it is selected. In FIG. 5, the **On Now** button 504 is highlighted, and if selected by actuating the **select (✓)** button 64, causes a third level display shown in FIG. 6 to appear and a further sorting an/or filtering of the data stored within RAM 40 (shown in FIG. 2).

Referring now to FIG. 6, display 600 shows what is on at the present time, which in this illustration is 6:30 p.m. A reduced representation 602 of all television shows that are on at the present time appears in FIG. 6. The reduced representation 602 presents each program that is presently on as a card in a tightly cascaded set of cards. The cards may be gray shade coded to distinguish between news shows, sport shows, dramatic shows, comedy shows, documentary shows and so forth. Those skilled in the art will recognize that color would be preferable for color television sets, and a method and apparatus according to the present invention using color to differentiated program types in the reduced representation 602 is contemplated. Thus, using visual coding within the reduced representation 602 would allow a sports program to visually stand out from the non-sports TV programming in the example shown. Up arrow 52 and Down arrow 54 respectively move a selection window 604, which is slightly wider than the items displayed in reduced representation, up and down the reduced representation 602 of the **On Now** subgroup in single steps. Motion of the active area along the reduced representation 602 is one dimensional, either up or down. The up arrows 60 and the down arrows 62 move this selection window 604 respectively up and down the reduced representation 602 in increments of six. The individual items visible and located within the selection window 604 represent a further subgroup of six programs out of the reduced representation 602 **On Now** subgroup. This six program subgroup of the selection window 604 is displayed in larger form in a grid display 606 located next to reduced representation 602. This larger form allows the viewer to read the titles of the programs presently in grid display 606. The visible coding, i.e. gray shade coding or color coding, of each item is retained in the larger form in grid display 606 to aid the viewer differentiate between the various types of programming offered.

Within selection window 604 and grid display 606 are active areas 605, 607 that highlight one item in their respective portions of display 600. The active areas 605, 607 move in coordination with each other in response to the Up arrow 52 and the Down arrow 54. When Up arrow 52 or Down arrow 54 require the active areas 605 and 607 to move above or below the selection window 604 and grid display 606, a paging occurs which moves the selection window up six or down six. When an item is located within active areas 605, 607, further information, such as the TV channel call sign, the cable channel number, and the exact start and stop times, is retrieved from the programming database stored in RAM 40 and displayed in the top of a frame 610 of display 600. If the **select (✓)** button 64 is actuated at this time, a preview of either a short text description or a brief still or motion video replaces the grid display 606. The data for these previews are stored in RAM 40. A second sequential actuation of the **select (✓)** button 64 actually selects the highlighted program in the active area 604 of reduced representation 602 and formerly highlighted in grid display 606. If the up arrow 52 or the down arrow 54 is actuated the respective preview for the next program item up or down from the previous previewed item is selected. The information displayed in the top of the frame 610 will change to the next program item up or down also. Actuation of the **cancel (X)** button 66 returns the viewer to the previous arrangement of display 600. The bottom of the frame 610 lists the characteristics of the display 600, which are also retrieved from RAM 40. If the **query (?)** button 65 is actuated, the grid display 606 will be replaced by a generalized help menu. This generalized help menu has many buttons, as explained below, one of which is a view button. If the view button is actuated, the generalized help menu is replaced with the previous **select (i.e. filter)** view.

Referring now to FIGs. 3, and 7 a selection of a program by category will be described. Actuation of the **query (?)** button 65 of controller 20 causes display 700 to appear on the screen of TV 10 (shown in FIG. 1). On display 700 has a help button 702, a categories button 704, a view button 706, a begin button 708, a favorites button 710, and a user

in a reasonable amount of time, so further filtering, either by a shorter time period, i.e. **On Now**, or a narrower category, i.e. basketball, is needed. To change to a narrower category, the viewer presses the query (?) button 62 which causes display 700 (shown in FIG. 7) to be displayed. Next, categories button 704 is selected which causes display 900 (shown in FIG. 9) to be displayed. Next, basketball button 903 is selected which causes display 1500 of FIG. 15 to be displayed.

5 The **Coming Up** time filter of FIGs. 12 and 13 has not been changed, so display 1500 shows the basketball programs coming up in the next 12 hours. As can be seen, the two-dimensional grid display 1500 contains approximately sixteen programs, which is sufficiently small to review each item individually in a reasonable time period. Moving active area 1502 around two-dimensional grid display 1500 with the up and down arrows 52, 54 and/or the right and left arrows 56, 58, causes the title and channel of each program to be displayed in the top of the frame of display 1500 to assist 10 the reviewing and selection process. For example, the program highlighted by active area 1502 is "This Week In the NBA" and it is showing on CNN. Thus, by selective filtering the unwieldy display 1300 of programs shown in FIG. 13 is reduced to a manageable handful of display 1500, which the viewer can navigate through individually in a reasonable time.

15 Referring now to FIGs. 16-23, another aspect of the present invention will be described. In FIG. 16 and the remaining figures, a longer period of time is selected other than the one and a half hours or so retrieved by the **On Now** selection. For example, if the viewer wishes to look at the programming available for the rest of the week in order to select something to record on a VCR (not shown). Actuating the button having the number zero (0) of the keypad 50 while watching a program causes the data view menu selection card, such as 900 of FIG. 9, to appear at the point in 20 the menu-display hierarchy where the last selection was made. Actuating the zero (0) button again moves the viewer towards the broadest data view menu 400 of FIG. 4, and the viewer may stop at any display in order to change time or subject matter categories.

25 Thus if a viewer were watching *This Week in the NBA*, and wanted to find a program of interest that is on later, the viewer would first actuate the zero (0) button of keypad 50 which would bring up the display of FIG. 10. Actuating the zero (0) button four more times takes the viewer through displays 900, 800, 700 and 500 of FIGs. 9, 8, 7 and 5 respectively. To get a specific program title, the search button 509 is actuated, which causes FIG. 16 to logically overlay the display 500. FIG. 16 shows a first display 1600 of an interactive alphanumeric selection sequence. First, all alphabetic titles are sorted into groups of five or less. If, for example, *Nova* was the title of the desired program, the active area would be moved from its initial position (either at the top of the display or at the last group selected) to the group of letters containing the letter N using the up arrow 52 or the down arrow 54 as shown in FIG. 17 followed by actuation 30 of the **select (')** button 64. This sequence would cause FIG. 18 to logically overlay FIG. 17. In FIG. 18, the active area is moved from its initial location at M to the location of N as shown in FIG. 17 followed again by actuation of the **select (')** button 64 causes the display 2000 of FIG. 20 to overlay FIG. 19. In display 2000 are single instances of the first 35 two letters, such as NYPD Blue is the only instance of N followed by Y, and multiple instances of the two letter string as denoted by the double right pointing arrows by NO. To continue the search for *Nova*, the active area is moved to the line containing NO of display 2000 as shown in FIG. 21 using the down arrow 56 and actuating the **select (')** button 64, which causes display 2200 of FIG. 22 to overlay display 2000. Now, *Nova* is the only instance of a program 40 beginning with NOV, so the entire title *Nova* appears in FIG. 22. By moving the active area to the line labeled NOVA in display 2200 and actuating the **select (')** button 64 causes the display 2300 shown in FIG. 23 to overlay display 2200 with a schedule of times and channels for the program series *Nova*.

45 FIG. 23 is a one week schedule that is laid out as a logical three dimensional grid. The days of the week are displayed along one side, in this case vertically along the left side, of the display 2300. Time of day is displayed along a perpendicular side, in this case horizontally across the top, for a twenty-four hour period. Thus, if an episode of *Nova* is scheduled at 8:00 p.m. on Sunday, a box of contrasting shade will be located in the intersection of the Sunday row and in the 8:00 p.m. column. The active area 2302 can be moved horizontally by arrows 56, 58 and vertically by arrows 52, 54 of keypad 50. If there are multiple occurrences of *Nova* on a particular night at a particular time, that fact is shown by a box, located at the intersection of the row of that day and the column of that time, having an asterisk (*) 50 located in the box. The asterisk (*) indicates the presence of a logical stack of multiple programs of *Nova* appearing on competing channels, such as occurs on Wednesday night at 8:00 p.m. To move or navigate through a stack of programs (or stack of episodes of programs with the same name, for example) on a particular day at a particular time slot, the viewer uses the double up arrows button 60 and the double down arrows button 62 for this third degree of freedom. Because the display 2300 may require greater visual discrimination than program title as a matter of course, the frame information window 1904 is larger than usual for display 2300. Further, frame 2304 is annotated with arrows indicating the existence of program episodes above or below the active areas' position in the stack. If the cable 16 has access to 300 plus 'channels' of programming, it is conceivable that some programs, such as *Nova* will be offered by 55 more than one channel at the same time. As described previously, once the viewer has moved the active area to a particular entry in two or three dimensions and actuates the **select (')** button 64, a selection is made. In this case, the selection sets an alarm to record a specific channel at a specific time at some day in the near future.

Referring back to FIGs. 1 and 2, overall operation of the apparatus of the invention is described. Program schedule

```

'===== COMING form code =====
'This form displays a TV schedule for several hours of one day.
'This version uses drawing methods for the program shapes
'  'as opposed to creating a control shape for each program)
'  and "point & shoot" or "visually closest" navigation.

5      Option Explicit
Dim allData(8) As snapshot 'all data within time period
Dim filterData(8) As snapshot 'a snapshot for each day in the view
Dim NDays As Integer 'number of days in display
Dim NSlots As Integer 'number of time slots in display
Dim NStation As Integer 'number of stations in display
Dim MaxStation As Integer 'total number of stations in database
Dim colorField As String 'the database field that determines item color
15     ' (the field should contain an integer)
Dim inPreview As Integer 'boolean 'should the preview message show?
Const sideGap = .05 'space at beginning and end of program
Const topGap = .4 'space btwn time label and first program shape
Dim refDate 'reference date for data time slots
Const lblHeight = 40 'height of day and time labels (in 500 scale)
Const MINProgWidth = .2 'minimum width of a program shape as fraction of slot
Dim slotsPerDay As Integer 'number of slots allowed per day
Dim currDay 'number of current day
Dim startTime 'start day and time of display

20     Dim TSBegin As Long 'first time slot
Dim TSEnd As Long 'last time slot
Dim TScurrent As Long 'current time slot
Dim rowOffset 'distance between (tops of) rows in the schedule

25     Sub ApplyFilter ()
'filter program data, keeping only the programs that match the query in filters(TV)
'also makes sure the number of stations is correct
'and the DB field determining the color is set
        Dim i As Integer 'counter

30     If InStr(filters(currDomain), "Station") Then
            NStation = 10 'note: this probably should be a variable or const, not 10
            colorField = "Type"
        Else
            NStation = MaxStation
            colorField = "Category"
        End If
        For i = 1 To NDays
            allData(i).Filter = filters(currDomain)
            Set filterData(i) = allData(i).CreateSnapshot()
        Next i
45     End Sub

Sub ChangeSel (d As String)
'Performs the navigation according to the direction parameter
        Dim current, firstMatch 'database markers
        Dim success As Integer 'boolean
        Dim s As Integer 'station
50

```

```

10 If Not success Then
11   'check all to left for "closest"
12   F.MoveFirst
13   While Not F.EOF
14     If F("StartTS") < TScurrent Then
15       dist = VDistHoriz(F("Station"), F("FinishTS"), s, TScurrent)
16       If dist < best Then
17         'keep best so far
18         best = dist
19         success = True
20         bestMark = F.Bookmark
21       End If
22     End If
23     F.MoveNext
24   Wend
25   If success Then
26     'move to best one
27     F.Bookmark = bestMark
28     TS = F("StartTS")
29   End If
30   End If
31 ElseIf d = "Down" Then
32   'check all programs below current one, keeping "closest"
33   While Not F.EOF
34     If F("Station") > s Then
35       dist = VDistVert(s, TScurrent, e, F("Station"), F("StartTS"),
36                           F("FinishTS"))
37       If dist < best Then
38         best = dist
39         success = True
40         bestMark = F.Bookmark
41       End If
42     End If
43     F.MoveNext
44   Wend
45   If success Then
46     F.Bookmark = bestMark
47     TS = F("StartTS")
48   End If
49 ElseIf d = "Up" Then
50   'check all programs above current one, keeping "closest"
51   While Not F.EOF
52     If F("Station") < s Then
53       dist = VDistVert(s, TScurrent, e, F("Station"), F("StartTS"),
54                           F("FinishTS"))
55       If dist < best Then
56         best = dist
57         success = True
58         bestMark = F.Bookmark
59       End If
60     End If
61     F.MovePrevious
62   Wend
63   If success Then

```

```

    popup.Top = lblTime(1).Top + 2 * lblTime(1).Height
    popup.Left = 2
    popup.Width = slotsPerDay - 3
    5      popup.Visible = True
    inPreview = True
    End Sub

    Sub DoSelect ()
    'set selection info and go to TV
    10     userStation = filterData(currDay)(“Station”)
    userStart = filterData(currDay)(“Start”)
    returnCode = TOTV
    Me.Hide
    End Sub

    15     Sub DrawProg (colorIndex, start, finish, station)
    'use drawing methods to draw a program shape
    'note: form.AutoRedraw should be set to true so the drawings are persistant
    Dim L, R, t, B  'left, right, top, bottom
    20     Dim dayStart
    Dim edge

    'convert a day/time to position in NSlot scale
    dayStart = startTime + currDay - 1
    L = (start - dayStart) * 48
    R = (finish - dayStart) * 48
    'clip shapes off at day boundaries
    If L < 0 Then L = 0
    If R > slotsPerDay Then R = slotsPerDay
    'place in correct day, with small gap between programs
    edge = (currDay - 1) * slotsPerDay
    25     L = L - edge + sideGap
    R = R - edge - sideGap
    'correct for min width to make sure program will show up
    If R - L < MINProgWidth Then R = L + MINProgWidth
    'set top according to station
    'note: this trick will not work if “favorite stations” are not numbered 1..n
    30     rowOffset = ((500 - 2 * lblHeight - shpProg(0).Height) / Nstation)
    t = shpSlot(0).Top + topGAP + (station - 1) * rowOffset
    B = t - shpProg(0).Height
    'draw the box with the correct color
    drawWidth = 1
    35     Me.FillStyle = 0 ‘solid
    Me.FillColor = Color(colorIndex Mod 9)
    Line (L, t)-Step(R - L, B - t), , B  'the line command with argument B draws a
    box
    End Sub

    40     Sub Form_Activate ()
    'make necessary changes to display, reset info and status bars
    Dim i As Integer 'counter
    Static saveFilter As String
    45
    50     If saveFilter = filters(currDomain) Then sameFilter = True

```

55

```

        returnCode = SHORTCUT
        Me.Hide
    End Select
    5
    'in any case
    If inPreview Then
        DoPreview
    Else
        popup.Visible = False
    10
    End If
End Sub

Sub Form_Load ()
    Dim i As Integer
    15
    Dim t 'as time

    'set form colors and fonts
    Me.BackColor = formCOLOR
    shpProg(0).BackColor = BorderColor
    20
    lblDay(0).BackColor = backgroundCOLOR
    lblAM.BackColor = backgroundCOLOR
    lblPM.BackColor = backgroundCOLOR
    selector.BorderColor = BorderColor
    dayLine(0).BorderColor = divideColor
    lblTime(0).ForeColor = slotCOLOR
    25
    shpSlot(0).BorderColor = slotCOLOR
    If displayMode = "TV" Then
        lblDay(0).FontSize = smallFONT
        lblTime(0).FontSize = smallFONT
        30
        lblAM.FontSize = smallFONT
        lblPM.FontSize = smallFONT
        popup.FontSize = mediumFONT
    Else
        35
        lblDay(0).FontSize = largeFONT
        lblTime(0).FontSize = largeFONT
        lblAM.FontSize = largeFONT
        lblPM.FontSize = largeFONT
        popup.FontSize = largeFONT
    End If
    'set scale and size objects
    SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
    40
    Me.Scale (0, 0)-(500, 500)
    SizeAControl lblDay(0), 0, lblHeight, 0, 500
    'note: the AM/PM labels would be placed when time is filtered
    SizeAControl lblPM, 0, lblHeight, 0, 30
    SizeAControl lblAM, 0, lblHeight, 500 - 30, 30
    SizeAControl lblTime(0), lblHeight, lblHeight, 0, 50
    45
    SizeAControl shpSlot(0), 2 * lblHeight + .5 * topGAP, 500 - 2 * lblHeight, 0, 50
    SizeAControl popup, 250, 200, 250, 200
    selector.BorderWidth = 1
    dayLine(0).Y1 = 0
    dayLine(0).Y2 = 500
    'initialize variables
    50

```

55

```

    End Sub

    Sub MakeDisplay ()
5     'create the display of programs from the data
        Dim i As Integer 'counter
        Dim d As Integer 'day
        Dim F As snapshot  'convenience

10    If Not sameView Then
        'would need to reset captions for times and day
        End If

15    'place program shapes
        Cls 'clear the form of previous drawings
        DoEvents 'make it so
        For d = 1 To NDays
            currDay = d
            'draw lines to separate time slots
            For i = 0 To slotsPerDay
                drawwidth = 4
20                Line (i, shpSlot(0).Top)-(i, 500), slotCCLR
                Next i
                'draw program shape for each program in data
                Set F = filterData(d)
                If Not F.EOF Then
25                    F.MoveFirst
                    Do While Not F.EOF
                        DrawProg F(colorField), F("Start"), F("Finish"), F("Station")
                        F.MoveNext
                    Loop
                    F.MoveFirst
30                End If
                Next d

35    'initialize stuff
        TScurrent = TSBEGIN
        currDay = 1
        shpProg(0).ZOrder
        selector.ZOrder
        Set F = filterData(currDay)
        'find a program to start on
        Do While TScurrent <= TSEnd
40            F.FindFirst Overlap(TScurrent, TScurrent)
            If Not F.NoMatch Then
                DisplayProg
                Exit Do
            End If
45            TScurrent = TScurrent + 1
        Loop
        'make sure TScurrent is in range
        If TScurrent > TSEnd Then TScurrent = TSBEGIN
    End Sub

50    Sub Position (shape As Control, start, finish, station)

```

55

```

10
11      If start1 > finish2 Then
12          deltaT = Abs(start1 - finish2)
13      ElseIf start2 > finish1 Then
14          deltaT = Abs(start2 - finish1)
15      Else
16          deltaT = 0
17      End If
18      VDistVert = deltaR + 2 * deltaT
19  End Function

20
21  ===== FRAME form code =====
22  'This form owns the standard info and status bars and allows
23  'transfer of control from form to form.
24  Option Explicit

25
26  Sub Form_Activate ()
27  'decides which other form should show in its display area
28      Select Case returnCode
29      Case SHOWVIEW
30          views(currDomain).Show
31      Case PICK
32          frmSelect.Show
33      Case TOTV
34          frmTV.Show
35      Case LASTVIEW
36          sameFilter = True
37          views(currDomain).Show
38      Case STARTUP
39          'do nothing--don't want rolodex to show yet
40      Case Else
41          frmDex.Show
42      End Select
43  End Sub

44
45  Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
46      If KeyCode = Asc("Q") Then
47          End
48      End If
49  End Sub

50
51  Sub Form_Load ()
52  'set colors and fonts
53      Me.BackColor = formCOLOR
54      sspInfo.FontSize = mediumFONT
55      sspStatus.FontSize = mediumFONT
56  'use builtin object to size background
57      ScrWidth = Screen.Width
58      ScrHeight = Screen.Height
59      If displayMode = "mini" Then
60          'for taking screen prints
61          ScrHeight = ScrHeight * .54
62          ScrWidth = ScrWidth * .712
63          displayMode = "TV"
64      Else
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
796
797
798
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
878
879
879
880
881
882
883
884
885
886
887
887
888
889
889
890
891
892
893
894
895
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
908
909
909
910
911
912
913
914
915
916
917
917
918
918
919
919
920
921
922
923
924
925
926
927
927
928
928
929
929
930
931
932
933
934
935
936
937
937
938
938
939
939
940
941
942
943
944
945
945
946
946
947
947
948
948
949
949
950
951
952
953
954
955
956
956
957
957
958
958
959
959
960
961
962
963
964
965
965
966
966
967
967
968
968
969
969
970
971
972
973
974
975
975
976
976
977
977
978
978
979
979
980
981
982
983
984
985
985
986
986
987
987
988
988
989
989
990
991
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
1641
1641
1642
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
1648
1648
1649
1649
1650
1650
1651
1651
1652
1652
1653
1653
1654
1654
1655
1655
1656
1656
1657
1657
1658
1658
1659
1659
1660
1660
1661
1661
1662
1662
1663
1663
1664
1664
1665
16
```

```

' database snapshots
Dim allData As snapshot
Dim itemData As snapshot
Dim storeData As snapshot
Dim deptData As snapshot
Dim stuffData As snapshot
Dim filterData As snapshot
Dim marker(1000) As String 'bookmarks of each MAXDISPLAY items
Dim locStart(1000) 'rItem index for start of locator

Sub ApplyFilter ()
  'filter the data according to user choice
  Dim sortString As String
  If Me Is TVlist Then
    captionField = "Title"
    sortString = ""
    colorField = "Type"
  ElseIf Me Is MOVlist Then
    If sameView Then
      'keep allData as it is
    Else
      'reset allData to all movies
      LoadData
      allData.Filter = viewFilter
      Set allData = allData.CreateSnapshot()
    End If
    captionField = "Title"
    sortString = "Title"
    colorField = "Type"
  ElseIf Me Is SHOPlist Then
    'note: This would all be done totally differently. Don't bother
    'understanding it, just rewrite it.
    Select Case filters(currDomain)
    Case "store"
      browsing = "store"
      Set allData = storeData
      captionField = "name"
      filters(currDomain) = ""
      sortString = "name"
      colorField = ""
    Case "dept"
      browsing = "dept"
      deptData.FindFirst "name = '" & userString & "'"
      userString = "" 'fix--this is cheating, I shouldn't use userString
      If deptData.NoMatch Then
        Set allData = deptData
        filters(currDomain) = ""
      Else
        browsing = "stuff"
        filters(currDomain) = "[dept code] = " & deptData("code")
        Set allData = stuffData
      End If
    End Select
  End If
End Sub

```

55

```

      Case "Up"
        If locSelected > 1 Then
          locSelected = locSelected - 1
          RedoDisplay
        End If
      Case "Down"
        If locSelected < MAXLOC Then
          locSelected = locSelected + 1
          RedoDisplay
        End If
      End Select
    End Sub

    Sub ChangeSel (direct As String)
      'navigate up or down one selection
      Select Case direct
        Case "Up"
          If itemSelected > 1 Then
            'move up within current display
            itemSelected = itemSelected - 1
            selector.Top = itemBox(itemSelected).Top - GAP
            rItem(0).Top = rItem(whichrItem(itemSelected)).Top
            rItem(0).Left = locL - GAP
            rItem(0).Width = locW + 2 * GAP
            SetItemInfo
          ElseIf locSelected > 1 Then
            'display previous section of list
            itemSelected = MAXDISPLAY
            locSelected = locSelected - 1
            RedoDisplay
          End If
        Case "Down"
          If itemSelected < MAXDISPLAY Then
            'move down within current display
            'do not move to select an empty item
            If (locSelected - 1) * MAXDISPLAY + itemSelected < MAXITEM Then
              itemSelected = itemSelected + 1
              selector.Top = itemBox(itemSelected).Top - GAP
              rItem(0).Top = rItem(whichrItem(itemSelected)).Top
              rItem(0).Left = locL - GAP
              rItem(0).Width = locW + 2 * GAP
              SetItemInfo
            End If
            ElseIf locSelected < MAXLOC Then
              'display next section of list
              itemSelected = 1
              locSelected = locSelected + 1
              RedoDisplay
            End If
          End Select
          rItem(0).Visible = True
        End Sub

        Sub DoPreview ()

```

55

```

Sub EndPreview()
  'go back to regular list operation
  Dim i As Integer 'counter
  5
  previewWin.Visible = False
  inPreview = False
  locator.Visible = True
  selector.Visible = True
  previewWin.Top = displayList.Top
  10
  RedoDisplay
End Sub

Sub Form_Activate()
  Dim i As Integer 'counter
  15
  Dim section As Integer 'count the number of locator locations
  Dim NVisible As Integer 'tally the visible shapes in a section
  Dim msg As String
  Static saveFilter As String
  Static saveView As String
  20
  'check new filters against current filters
  If Not sameView Then sameView = (saveView = viewFilter)
  saveView = viewFilter
  If Not sameFilter Then sameFilter = (saveFilter = filters(currDomain))
  saveFilter = filters(currDomain)
  25
  SetStatus currView(currDomain) & currFilter(currDomain), greyCOLOR
  If sameFilter And sameView Then
    'keep everything the same as last time
    If newUser And Not Me Is SHOPlist Then
      30
      Popup.Caption = "To change the category shown, press the 'Category' button.."
      popup.Visible = True
      newUser = False
    End If
    35
    RedoDisplay
  Else
    'clear up display
    SetInfo "Selecting data, please wait...", GREY
    If MAXITEM = 0 Then
      40
      previewWin.Caption = ""
      previewWin.Visible = False
    End If
    DoEvents
    If inPreview Then EndPreview
    45
    For i = 1 To MAXDISPLAY
      itemBox(i).Caption = ""
    Next i
    For i = 1 To MAXITEM
      50
      Unload xItem(i)
    Next i
  End If
End Sub

```

55

```

        If filterData("StartTS") < TS Then
            rItem(i).Left = rItem(i).Left - reducedEXTRA
            rItem(i).Width = rItem(i).Width + reducedEXTRA
        End If
        If filterData("FinishTS") > TS Then
            rItem(i).Width = rItem(i).Width - reducedEXTRA
        End If
    End If
    rItem(i).ZOrder
    rItem(i).Visible = True
    filterData.MoveNext
    Next i
    MAXLOC = section
    locStart(section + 1) = MAXITEM + 1
    15
    'set length of minselector (use rItem(0))
    rItem(0).Left = locL - GAP
    rItem(0).Width = locW + 2 * GAP

    'initialize selector and locator
    20
    itemSelected = 1
    locSelected = 1
    locator.Visible = True
    rItem(0).BackColor = highlightCOLOR
    'set the captions in the itemBoxes
    25
    RedoDisplay
End If
End If
End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    30
    popup.Visible = False
    Select Case KeyCode
        Case Asc("Q")
            End
        Case B_BACK
            35
            If Me Is SHOPlist And browsing = "item" Then
                'not exactly what we want
                returnCode = ALPHA
                Me.Hide
            Else
                40
                returnCode = BACK
                Me.Hide
            End If
        Case B_HELP
            InvokeHelp
        Case B_PREVIEW
            45
            If inPreview Then
                EndPreview
            Else
                DoPreview
            End If
        Case B_SELECT
            50
            If MAXITEM > 0 Then DoSelect
    End Select
End Sub

```

55

```

        previewWin.FontSize = mediumFONT
        popup.FontSize = smallFONT
    End If
    5    rItem(0).BackColor = itemCOLOR
    selector.FillColor = highlightCOLOR
    displayList.FillColor = backgroundCOLOR
    previewWin.BackColor = backgroundCOLOR
    locator.FillColor = backgroundCOLOR
    itemBox(0).BackColor = itemCOLOR
    leftArrow(0).BackColor = itemCOLOR
    rightArrow(0).BackColor = itemCOLOR
    shpSlot.BorderColor = slotCOLOR
    'size the objects to the screen
    SizeForm Me, DispTop, DispHeight, DispLeft, DispWidth
    10   Me.Scale (0, 0)-(1000, 1000)
    SizeAControl locator, T - GAP, H + GAP, locL - GAP, locW + 2 * GAP
    SizeAControl shpSlot, T, H, locL - reducedEXTRA, locW - 2 * reducedEXTRA
    SizeAControl displayList, T - GAP, H - GAP, dispL, dispW
    SizeAControl popup, dispW / 2, 4 * locW, dispW / 2, 4 * locW
    CPlace 1, previewWin, displayList
    15   locator.ZOrder
    shpSlot.ZOrder
    rItem(0).ZOrder
    itemRoom = H / MAXDISPLAY
    SizeAControl itemBox(0), T + (.5 * GAP), itemRoom - GAP, dispL + EXTRA, dispW -
    20   - EXTRA
    SizeAControl patch(0), 50, (6.8 * itemBox(0).Height), (12.3 * itemBox(0).Width),
    25   (7 * itemBox(0).Height)
    If displayMode = "TV" Then
        patch(0).Left = 8.08 * itemBox(0).Width
        patch(0).Height = 3.7 * itemBox(0).Height
    30   End If
    SizeAControl leftArrow(0), T + (.5 * GAP), itemRoom - GAP, dispL, EXTRA
    SizeAControl rightArrow(0), T + (.5 * GAP), itemRoom - GAP, dispL + dispW -
    EXTRA, EXTRA
    35   SizeAControl selector, T, itemRoom + GAP, dispL, dispW
    selector.ZOrder
    For i = 1 To MAXDISPLAY
        'Load itemBox(i) 'Now created at design time--fixed number (6)
        itemBox(i).Visible = False
        40   CCopy itemBox(0), itemBox(i)
        patch(i).Visible = False
        CPlace 0, patch(i), patch(0)
        itemBox(i).Top = itemBox(0).Top + (i - 1) * itemRoom
        Load leftArrow(i)
        leftArrow(i).Top = itemBox(i).Top
        Load rightArrow(i)
        rightArrow(i).Top = itemBox(i).Top
    45   Next i

    'load the list data and set up the display
    sameFilter = False
    sameView = False
    50   LoadData

```

55

```

      'set the captions in the itemBoxes to correspond to items in locator
      'reposition locator and selector, update info box

5       Dim last As Integer
      Dim i As Integer
      Dim Index As Integer 'index of rItem

10      If MAXITEM = 0 Then Exit Sub
      'figure first item location
      filterData.Bookmark = marker(locSelected)

15      Index = locStart(locSelected)
      For i = 1 To MAXDISPLAY
          If filterData.EOF Then
              'hide empty itemBox
              itemBox(i).Caption = ""
              itemBox(i).Visible = False
              leftArrow(i).Visible = False
              rightArrow(i).Visible = False
          Else
              whichrItem(i) = Index 'so we can highlight the correct rItem (reduced
item)
              If colorField <> "" Then itemBox(i).BackColor =
Color(filterData(colorField) Mod 9)
              itemBox(i).Caption = filterData(captionField)
              If Not inPreview Then itemBox(i).Visible = True
25          If Me Is TVlist And Not inPreview Then
              'show arrows to reflect program length
              If filterData("StartTS") < TS Then
                  leftArrow(i).BackColor = itemBox(i).BackColor
                  leftArrow(i).Visible = True
30          Else
                  leftArrow(i).Visible = False
              End If
              If filterData("FinishTS") > TS Then
                  rightArrow(i).BackColor = itemBox(i).BackColor
                  rightArrow(i).Visible = True
35          Else
                  rightArrow(i).Visible = False
              End If
              'show color patch for subcategory
              patch(i).FillColor = Color(filterData("Category") Mod 9)
              patch(i).Visible = True
40          End If
              last = i
              Index = Index - 1
              filterData.MoveNext
45          End If
      Next i

      'Do not allow blank to be selected
      If itemSelected > last Then
          itemSelected = last
50      End If

```

55

```

'===== MESSAGE form code =====
'This form is used by Help and some lists to display information.
' temporarily covering up the current form.
Option Explicit

Const GAP = 500

10 Sub Form_Activate ()
    textArea.Caption = userMsg
End Sub

15 Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
    Case Else
        returnCode = KeyCode
        Me.Hide
    End Select
End Sub

20 Sub Form_Load ()
    'set colors and fonts
    Me.BackColor = itemCOLOR
    textArea.BackColor = itemCOLOR
    textArea.FontSize = largeFONT
    'set sizes
    SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
    SizeAControl textArea, GAP, DispHeight - 2 * GAP, GAP, DispWidth - 2 * GAP
    'initialize
    textArea.Caption = ""
End Sub

25 '===== ROLODEX form code =====
'This form shows the main menu and filter menus.
'Unimplemented: Have filter button color correspond to type/category color
Option Explicit

30 Dim BlinkControl As Control 'pointer to blinking highlight
Dim parent As Integer 'number of parent card
Dim current As Integer 'number of current card

35 'special cards
'note: these must be updated each time the number of filter cards in the card
datafile changes
Const filterCARD = 1 'TV filter menu
Const movieCARD = 68 'movie filter menu
Const homeCARD = 96 'main menu
40 Dim lastCard As Integer 'holds number of regular card while in filter

45 Const MAXTITLE = 3 'WARNING: A change in MAXTITLE requires a change in code for
LoadGraphics
Const CARDSHIFT = 2.5 'for card display--amount change in card placement
Const MAXROWS = 3 'for card display--number of rows of buttons
Const MAXCOLS = 3 'for card display--number of columns of buttons on a card

```

55

```

      sspCont.BackColor = sspCard(0).BackColor
      Zoom 10, sspCont, sspCard(depth)
      DisplayCard cardNo
      sspCont.Visible = False
      End If
      Case "Select"
      index = Cards(current).selected
      If index > 0 Then
      CCopy sspItem(index), sspCont
      sspCont.Visible = True
      sspCont.BackColor = sspCard(0).BackColor
      SizeACcontrol sspCard(0), 0, 500, 0, 500 'size of whole form
      Zoom 10, sspCont, sspCard(0)
      End If
      End Select
End Sub

Sub BlinkStart (C As Control, vis)
'enable blinking object
  Set BlinkControl = C
  BlinkControl.Visible = vis
  tmrBlink.Enabled = True
End Sub

Sub BlinkStop (vis)
'stop blinking object, leaving visibility as vis
  tmrBlink.Enabled = False
  If BlinkControl Is Nothing Then
  'do nothing
  Else
  BlinkControl.Visible = vis
  End If
  Set BlinkControl = Nothing
End Sub

Sub ButtonAction ()
'perform action associated with selected button
  Dim button As Integer
  Dim cardNo As Integer
  Dim msg As String

  button = Cards(current).selected 'item number of selected button on parent card
  cardNo = Cards(current).item(button) 'card number of selected button
  If button < 1 Then Exit Sub

  Select Case Cards(cardNo).actionCode
  Case actNONE
  'an inactive button
  SetInfo "This option is not yet available.", greyCOLOR
  Case actNEXT
  'display the next card
  Animate "Next", Cards(current).item(button)
  Case actDOMAIN
  'change current domain before going to the next card

```

55

```

5      Case actALPHASHOP
6          'get a string from user, search for items beginning with user string
7          'note: this would probably be very different
8          Animate "Select", 0
9          SetStatus "Shopping. " & Cards(cardNo).name, greyCOLOR
10         msg = Cards(cardNo).actionData
11         SetInfo msg, YELLOW
12         Wait frmAlpha
13         If returnCode <> BACK And userString <> "" Then
14             sameFilter = False
15             filters(currDomain) = "item"
16             Set views(currDomain) = listFrm(currDomain)
17             returnCode = SHOWVIEW
18             Me.Hide
19         End If
20     Case actALPHATV
21         'allow user to select a show title
22         Animate "Select", 0
23         SetStatus "TV, " & Cards(cardNo).name, greyCOLOR
24         returnCode = PICK
25         Me.Hide
26     Case actALPHAMOV
27         'This is not hooked up to work, but would probably be
28         'a lot like actALPHTV
29         Animate "Select", 0
30     Case actFILTER
31         'send a new filter to a TV view
32         filters(currDomain) = Cards(cardNo).actionData
33         currFilter(currDomain) = Cards(cardNo).infotext
34         sameFilter = False
35         sameView = True
36         returnCode = SHOWVIEW
37         Me.Hide
38     Case actMOVIE
39         'show a movie list
40         Animate "Select", 0
41         If current > homeCARD Then
42             'the view (a filter) is changing
43             currView(currDomain) = Cards(cardNo).infotext
44             viewFilter = Cards(cardNo).actionData
45             sameView = False
46             sameFilter = False
47         Else
48             'the category is changing
49             currFilter(currDomain) = ":" & Cards(cardNo).infotext
50             filters(currDomain) = Cards(cardNo).actionData
51             sameView = True
52             sameFilter = False
53         End If
54         Set views(currDomain) = listFrm(currDomain)
55         returnCode = SHOWVIEW
56         Me.Hide
57     Case actSTORE
58         'show a list of stores

```

55

```

      currDomain = "TV"
      filters(currDomain) = ""
      sameFilter = False
      sameView = True
      Set views(currDomain) = listFrm(currDomain)
      returnCode = SHOWVIEW
      Me.Hide
      Case actKEYS
        'Only for development, wouldn't stay
        SetKeys Cards(cardNo).actionData
        SetStatus Cards(cardNo).infotext, itemCOLOR
        current = homeCARD
        DisplayCard current
      Case actTABS
        'only for development
        ToggleTabs
      Case Else
        MsgBox "Bad action code for card " & Cards(cardNo).name
        Stop
        End
      End Select
    End Sub

    Sub ChangeSel (direct As String)
      'do button navigation
      Dim n As Integer
      Dim last As Integer, Sel As Integer
      n = Cards(current).NItems
      last = Cards(current).selected
      If last = 0 Then Exit Sub

      If direct = "Right" Then
        'move right with wrap around
        If last = n Then
          Sel = 1
        Else
          Sel = last + 1
        End If
      ElseIf direct = "Left" Then
        'move left with wrap around
        If last = 1 Then
          Sel = n
        Else
          Sel = last - 1
        End If
      ElseIf direct = "Up" Then
        'move up, no wrap around
        If last > MAXCOLS Then
          Sel = last - MAXCOLS
        Else
          Sel = last
        End If
      ElseIf direct = "Down" Then
        'move down, no wrap around
    End Sub
  
```

```

NItems = Cards(current).NItems

5      Set Area = sspCard(Cards(current).level)'this is a pointer, not a copy
'calculate size of button
Dx = Area.Width * .9 / MAXCOLUMNS
Dy = Area.Height * .9 / MAXROWS
w = Dx * .9
If w > 30 Then w = 30
h = Dy * .9
If h > 20 Then h = 20

sspBlinkBG.Visible = False
sspBlinkBG.ZOrder 0  'bring to front
'place and show each button
15     For i = 1 To NItems
        sspItem(i).Width = w
        sspItem(i).Height = h
        sspItem(i).Caption = Cards(Cards(current).item(i)).name
        If Cards(Cards(current).item(i)).actionCode = actNONE Then
            'turn inactive buttons grey
            sspItem(i).BackColor = greyCOLOR
        Else
            sspItem(i).BackColor = itemCOLOR
        End If
        25      x = Area.Left + .05 * Area.Width + ((i - 1) Mod MAXCOLUMNS) * .5 * Dx
        Y = Area.Top + .05 * Area.Height + (Int((i - 1) / MAXCOLUMNS) * .5) * Dy
        CenterItem sspItem(i), x, Y
        sspItem(i).ZOrder 0
        sspItem(i).Visible = True
    Next i
    'make blinker bigger than buttons
    CPlace 2, sspBlinkBG, sspItem(1)

    'hide unused buttons
    For i = NItems + 1 To MAXITEM
        sspItem(i).Visible = False
    Next i
    UpdateSel
End Sub

Sub Form_Activate ()
40      'check for a return code from another form
        sspCont.Visible = False
        Select Case returnCode
        Case BACK
            If current < homeCARD Then current = lastCard
            SetStatus "Use arrows and select or use keypad.", greyCOLOR
            DisplayCard current
            UpdateSel
        Case SHORTCUT
            current = homeCARD
            SetStatus "Use arrows and select or use keypad.", greyCOLOR
            DisplayCard current
50      End Sub

```

55

```

Case B_PAGEDOWN
  'use numeric key pad to choose a button directly, without navigation
Case B_1
  6   If Cards(current).NItems > 0 Then
      Cards(current).selected = 1
      UpdateSel
      ButtonAction
    End If
Case B_2
  10  If Cards(current).NItems > 1 Then
      Cards(current).selected = 2
      UpdateSel
      ButtonAction
    End If
Case B_3
  15  If Cards(current).NItems > 2 Then
      Cards(current).selected = 3
      UpdateSel
      ButtonAction
    End If
Case B_4
  20  If Cards(current).NItems > 3 Then
      Cards(current).selected = 4
      UpdateSel
      ButtonAction
    End If
Case B_5
  25  If Cards(current).NItems > 4 Then
      Cards(current).selected = 5
      UpdateSel
      ButtonAction
    End If
Case B_6
  30  If Cards(current).NItems > 5 Then
      Cards(current).selected = 6
      UpdateSel
      ButtonAction
    End If
Case B_7
  35  If Cards(current).NItems > 6 Then
      Cards(current).selected = 7
      UpdateSel
      ButtonAction
    End If
Case B_8
  40  If Cards(current).NItems > 7 Then
      Cards(current).selected = 8
      UpdateSel
      ButtonAction
    End If
Case B_9
  45  If Cards(current).NItems > 8 Then
      Cards(current).selected = 9
      UpdateSel
    End If
  50

```

55

```

        sspCard(i).Top = sspCard(i - 1).Top - CARDSHIFT
        sspCard(i).ZOrder
        Load sspTitle(i)
5       sspTitle(i).Top = sspCard(i).Top - sspTitle(0).Height + 2
        Select Case (i Mod MAXTITLE)
        'note: these cases are not flexible for different MAXTITLE
        Case 1
            sspTitle(i).Left = sspCard(i).Left
        Case 2
            sspTitle(i).Left = sspCard(i).Left + sspCard(i).Width / 2 -
        sspTitle(i).Width / 2
            Case 0
                sspTitle(i).Left = sspCard(i).Left + sspCard(i).Width - sspTitle(i).Width
        End Select
15      sspTitle(i).ZOrder
        Next i
    End Sub

    Sub PopulateCards ()
        'This subroutine reads in the card data from the
        'CARDFILE file defined as a constant. The cards
        'will be numbered 1 to the number of lines (cards)
        'in the file. All special cards should come before
        'the home card (by convention), and are named as
        'constants in the declarations. Each card record
        'should have a level (integer), item selected (integer),
        'a name (string), an info string (string), and
        'an action code (integer). If the action code is greater
        'than actNEXT, one additional input (variant type) is read
        'for the card.
30      Dim last As Integer, parent As Integer
        Dim selected As Integer
        Dim index As Integer, itemNo As Integer
        Dim level, title, text, action
        Open CARDFILE For Input As #1

35      'make dummy parent for top level
        index = 0
        Cards(index).name = "root"
        Cards(index).level = 0
        Cards(index).NItems = 0
        While Not EOF(1)
40      last = index
        index = index + 1
        Input #1, level, selected, title, text, action
        Cards(index).level = level
        Cards(index).selected = selected
45      Cards(index).name = title
        Cards(index).infctext = text
        Cards(index).actionCode = action
        If action > actNEXT Then
            Input #1, action
            Cards(index).actionData = action
50      End If
    End Sub

```

```

i = Cards(current).selected
If i > 0 Then 'something is selected
  Set S = sspItem(i) 'S is pointer to button
  'find center of button
  x = S.Left + S.Width / 2
  Y = S.Top + S.Height / 2
  'put blinker behind button
  CenterItem sspBlinkBG, x, Y
  'resume blinking
  BlinkStart sspBlinkBG, True
End If
text = Cards(Cards(current).item(Cards(current).selected)).infotext
color = sspItem(Cards(current).selected).BackColor
SetInfo text, color
End Sub

Sub Zoom (n As Integer, C As Control, Dest As Control)
'animates control C changing size to control Dest
  Dim i As Integer, j As Integer
  Dim dl, dw, dt, dh
  dl = (Dest.Left - C.Left) / n
  dw = (Dest.Width - C.Width) / n
  dt = (Dest.Top - C.Top) / n
  dh = (Dest.Height - C.Height) / n
  C.ZOrder
  C.AutoSize = False
  For i = 1 To n
    C.Move C.Left + dl, C.Top + dt, C.Width + dw, C.Height + dh
    C.Refresh
  Next i
End Sub

'===== SELECT form code =====
'This form is another attempt at alphabetic input that allows only valid input.
' It relies on the TV titles database which has two tables. The reference table is
used first
' and contains a count of all items starting with each letter of the alphabet or
with a
' symbol or number. The user is first presented with a list of possible starting
letters
' (each item in the first on-screen list may have several letters in it). Once a
starting
' letter is chosen, a snapshot is made of matching entries from the table of titles.
' Each list the user sees has only valid choices for the next letter, or full titles
if
' a particular title is distinguished from all others by the letters chosen so far.
' The best way to understand is to see the form in action before reading the code.
'The code could easily be modified to work with other data such as lists of movies,
etc.
'note: the non-proportional font used in the itemBoxes is Courier New
Option Explicit
Dim DB As database 'the full database

Dim list(1000) As String 'the list of selection strings

```

55

```

        If locSelected < MAXLOC Then
            locSelected = locSelected + 1
            RedoDisplay
5        End If
        End Select
    End Sub

    Sub ChangeSel (direct As String)
        'Perform list navigation
10       Select Case direct
        Case "Up"
            If itemSelected > 1 Then
                'move up within items currently displayed
                itemSelected = itemSelected - 1
15                selector.Top = itemBox(itemSelected).Top - GAP
                rItem(0).Top = locator.Top + rowOffset * (itemSelected - 1)
                SetItemInfo
            ElseIf locSelected > 1 Then
                'display previous section of the list
20                itemSelected = MAXDISPLAY
                locSelected = locSelected - 1
                RedoDisplay
            End If
        Case "Down"
            If itemSelected < MAXDISPLAY Then
                'move down within items currently displayed
                'do not move to select an empty item
                If (locSelected - 1) * MAXDISPLAY + itemSelected < MAXITEM Then
                    itemSelected = itemSelected + 1
                    selector.Top = itemBox(itemSelected).Top - GAP
30                    rItem(0).Top = locator.Top + rowOffset * (itemSelected - 1)
                    SetItemInfo
                End If
                ElseIf locSelected < MAXLOC Then
                    'display next section of list
                    itemSelected = 1
35                    locSelected = locSelected + 1
                    RedoDisplay
                End If
            End Select
        End Sub

40    Sub DoSelect ()
        'finish with leaf value or create a new list based on user's choice of prefix
        Dim index As Integer
        Dim count As Integer
        Dim i As Integer
45        Dim nextChar As String
        Dim looking As Integer 'boolean
        Dim title As String

        index = locStart(locSelected) + itemSelected - 1 'index in list of item
50        selected
        If leaf(index) Then

```

55

```

        End If

        'data assumed to be already sorted
        If Not initialList Then
            'still need to create new list from data
            Set filterData = filterData.CreateSnapshot()
            filterData.MoveFirst
            listEnd = 0
            For i = Asc(" ") To Asc("Z") 'space, punctuation, and letters
                'note: should be fixed up by not trying every single one, go straight to
                10 next db item's char
                    count = 0: looking = True
                    While Not filterData.EOF And looking
                        nextChar = Mid(filterData("SelectTitle"), Len(currPrefix) + 1,
                        15     1)
                        If nextChar = Chr(i) Or nextChar = LCase(Chr(i)) Then
                            count = count + 1
                            filterData.MoveNext
                        Else
                            looking = False
                        End If
                        20 Wend
                        Select Case count
                            Case 0 'do not add to list
                            Case 1 'make a leaf entry
                                filterData.MovePrevious
                                25 listEnd = listEnd + 1
                                list(listEnd) = fixAmpersand((filterData("SelectTitle")))
                                leaf(listEnd) = True
                                filterData.MoveNext
                            Case Else 'make a non-leaf entry
                                filterData.MovePrevious
                                30 listEnd = listEnd + 1
                                list(listEnd) = currPrefix & "=" & Chr(i) 'underline new char
                                'note: underlining is just one mechanism for emphasizing what is
                                different
                                leaf(listEnd) = False
                                35 filterData.MoveNext
                        End Select
                    Next i
                    If filterData.RecordCount <= MAXDISPLAY Then
                        'redo the list to have just leaves in it, if they all fit in one
                        40 display
                            listEnd = 0
                            filterData.MoveFirst
                            While Not filterData.EOF
                                45 listEnd = listEnd - 1
                                list(listEnd) = fixAmpersand((filterData("SelectTitle")))
                                leaf(listEnd) = True
                                filterData.MoveNext
                            Wend
                        End If
                    End If
        50

```

```

        Dim itemRoom

5         'set colors and fonts
        itemBox(0).FontSize = largeFONT
        rightArrow(0).FontSize = largeFONT
        rItem(0).BackColor = itemCOLOR
        selector.FillColor = highlightCOLOR
        displayList.FillColor = backgroundCOLOR
        locator.FillColor = backgroundCOLOR
10        itemBox(0).BackColor = itemCOLOR
        rightArrow(0).BackColor = itemCOLOR
        shpSlot.BorderColor = sletCOLOR
        'size and place the objects to the screen
        SizeAForm Me, DispTop, DispHeight, DispLeft, DispWidth
15        Me.Scale (0, 0)-(1000, 1000)
        SizeAControl locator, T - GAP, H + GAP, locL - GAP, locW + 2 * GAP
        SizeAControl shpSlot, T, H, locL + reducedEXTRA, locW - 2 * reducedEXTRA
        SizeAControl displayList, T - GAP, H + GAP, dispL, dispW
        locator.ZOrder
20        shpSlot.ZOrder
        rItem(0).ZOrder
        itemRoom = H / MAXDISPLAY
        SizeAControl itemBox(0), T + (.5 * GAP), itemRoom - GAP, dispL + EXTRA, dispW -
25        2 * EXTRA
        SizeAControl leftArrow(0), T + (.5 * GAP), itemRoom - GAP, dispL, EXTRA
        SizeAControl rightArrow(0), T + (.5 * GAP), itemRoom - GAP, dispL + dispW -
        EXTRA, EXTRA
        SizeAControl selector, T, itemRoom + GAP, dispL, dispW
        selector.ZOrder
        For i = 1 To MAXDISPLAY
30        Load itemBox(i)
        itemBox(i).Visible = False
        itemBox(i).Top = itemBox(0).Top + (i - 1) * itemRoom
        Load rightArrow(i)
        rightArrow(i).Top = itemBox(i).Top
        Next i
35        End Sub

        Sub LoadData ()
        Dim refSnap As snapshot
        Const MAXTOGETHER = MAXDISPLAY 'number of letter allowed in one itemBox
        Dim together
40

        'fill initial selection list
        listEnd = 0
        Set DB = OpenDatabase(TVTitles)
        Set allData = DB.CreateSnapshot("Titles")
45

        'create initial list
        Set refSnap = DB.CreateSnapshot("Reference")
        refSnap.MoveFirst
        together = MAXTOGETHER 'indicate need for new item
        While Not refSnap.EOF
50        Select Case refSnap("Number")

```

55

```

5      rItem(0).Visible = False
      rItem(0).Top = T
      rItem(0).Left = locL + reducedEXTRA
      rItem(0).Width = locW - 2 * reducedEXTRA
      rItem(0).BackColor = itemCOLOR
      filterData.MoveFirst
      'size and place the item shapes
      'and set section bookmarks
      section = 0      'number of locator locations
10     For i = 1 To MAXITEM
            Load rItem(i)
            rItem(i).Top = T + (i - 1) * rowOffset
            If ((i - 1) Mod MAXDISPLAY) = 0 Then
                'begin a new locator location
15             section = section + 1
                locStart(section) = i
            End If
            If Not leaf(i) Then
                rItem(i).Width = rItem(i).Width + reducedEXTRA
20             End If
                rItem(i).ZOrder
                rItem(i).Visible = True
            Next i
            MAXLOC = section
            locStart(section + 1) = MAXITEM + 1
25
            'set length of minselector
            'use rItem(0) as mini selector
            rItem(0).Left = locL - GAP
            rItem(0).Width = locW + 2 * GAP
30
            'initialize selector and locator
            itemSelected = 1
            locSelected = 1
            rItem(0).BackColor = highlightCOLOR
35
            'set the captions in the itemBoxes
            RedoDisplay
        End Sub

        Sub RedoDisplay ()
40        'set the captions in the itemBoxes to correspond to items in locator
        'reposition locator, selector and set item info in info box

            Dim last As Integer 'number of last item in display
            Dim i As Integer 'counter
            Dim index As Integer 'index of item in list
45
            index = locStart(locSelected)
            For i = 1 To MAXDISPLAY
                If index > MAXITEM Then
                    'hide empty itemBox
                    itemBox(i).Caption = ""
50                    itemBox(i).Visible = False

```

55

```

      index = locStart(locSelected) + itemSelected - 1
      If leaf(index) Then
        'get full title from data
        Set F = filterData
        F.FindFirst "SelectTitle = "'' & list(index) & "''"
        msg = F("FullTitle")
      Else
        msg = "Titles beginning with "'' & list(index) & "''"
      End If
      SetInfo msg, (itemBox(itemSelected).BackColor)
    End If
  End Sub

  Sub tmrBlink_Timer ()
    BlinkControl.Visible = Not BlinkControl.Visible
  End Sub

  '===== START form code =====
  'This startup form allows the developer to choose display mode
  ' (either for PC, TV, or mini PC for making screen prints;
  ' then starts the actual program by calling Main
  Option Explicit

  Sub Form_Load ()
    returnCode = STARTUP
  End Sub

  Sub miniButton_Click ()
    displayMode = "mini"
    Unload Me
    Main
  End Sub

  Sub PCbutton_Click ()
    displayMode = "PC"
    Unload Me
    Main
  End Sub

  Sub TVbutton_Click ()
    displayMode = "TV"
    Unload Me
    Main
  End Sub

  '===== TV form code =====
  'This form pretends to show a TV program or record it. if it is not currently on
  Option Explicit

  Const GAP = 700

  Sub Form_Activate ()
    Dim msg As String

```

55

```

    Case AscI("Q")
        End
    Case Else
        returnCode = BACK
        Me.Hide
    End Select
End Sub

5      Sub Form_Load ()
        textArea.Caption = ""
        textArea.FontSize = largeFONT
        SizeAForm Me, 0, ScrHeight, 0, ScrWidth
        SizeAControl textArea, GAP, ScrHeight - 2 * GAP, GAP, ScrWidth - 2 * GAP
End Sub

10     '===== TV_GUIDE form code =====
'General remarks:
'    The Main procedure starts the ball rolling by showing the Frame, loading
'    all the forms, and then showing the rolodex menu. Control is transferred from form
'    to form through the use of the returnCode variable (see list of return codes in
'    global declarations). The frmDex, for example, sets the returnCode to SHOWVIEW, and
'    hides itself. This causes frmFrame to become active. frmFrame looks at the
'    returnCode
'    and shows the current domain's view form. Communication between forms is done
'    through
'    a variety of variables, since a form's procedures are not accessible from outside.
25

        Option Explicit
'.....
'    Global Declarations
'.....
30      'database constants
Global Const CARDFILE = "c:\pctv\db\cards2.txt"
Global Const MVDB = "c:\pctv\db\plots.mdb"
Global Const SPDB = "c:\pctv\db\shopping.mdb"
Global Const TVDB = "c:\pctv\db\big.mdb"
Global Const TVTitles = "c:\pctv\db\titles.mdb"
35      Const CATDB = "c:\pctv\db\cats.mdb"
Dim typeTable As table 'TV type IDs
Dim catTable As table 'TV category IDs
Dim statTable As table 'station IDs
Global fakeToday 'keep the day constant
40      Global fakeTime 'keep the time constant
Global displayMode As String 'display set for "PC" or "TV" (affects size of fonts
and graphics)
Global newUser As Integer 'boolean 'when true, give extra helps

45      Global ScrWidth, ScrHeight
Global DispTop, DispHeight, DispLeft, DispWidth 'display area available to forms
inside the frame

'Colors
50      Global Const highlightCOLOR = &H8080FF 'reddish
Global Const backgroundCOLOR = &H80FFFF 'yellow

```

55

```

5      Global Const FILTER = 5
      Global Const COMING = 6
      Global Const SHOWVIEW = 9
      Global Const ALPHA = 10
      Global Const PICK = 11
      Global Const STARTUP = 12

      .....  

10     ' Define Type Card
      ' for rolodex
      .....  

      Global Const MAXITEM = 9  'max number of buttons on a card

      'Represents one index card as viewed on screen
15     Type Card
      self As Integer 'item number of self on parent
      level As Integer 'number of cards away from root
      name As String  'text to appear on button/card
      infotext As String 'text for info bar
      actionCode As Integer 'code for action to take when chosen
20     actionData As String 'extra info needed for action
      parent As Integer 'number of parent card
      NItems As Integer 'number of buttons visible on card
      Item(MAXITEM) As Integer 'array of card pointers (one for each button on card)
      selected As Integer 'the number of the selected button
25     End Type

      'Array of up to MAXCARDS index cards
      Global Const MAXCARDS = 1000
      Global Cards(MAXCARDS) As Card

30     .....  

      ' Remote Buttons
      .....  

      'assigned values in sub SetKeys
35     Global B_BACK
      Global B_HELP
      Global B_PREVIEW
      Global B_UP
      Global B_DOWN
      Global B_LEFT
      Global B_RIGHT
40     Global B_SELECT
      Global B_PAGEUP
      Global B_PAGEDOWN
      Global B_1
      Global B_2
45     Global B_3
      Global B_4
      Global B_5
      Global B_6
      Global B_7
      Global B_8
50     Global B_9

```

```

' KEY_A thru KEY_Z are the same as their ASCII equivalents: 'A' thru 'Z'
' KEY_0 thru KEY_9 are the same as their ASCII equivalents: '0' thru '9'

5   Global Const KEY_NUMPAD0 = &H60
    Global Const KEY_NUMPAD1 = &H61
    Global Const KEY_NUMPAD2 = &H62
    Global Const KEY_NUMPAD3 = &H63
    Global Const KEY_NUMPAD4 = &H64
10  Global Const KEY_NUMPAD5 = &H65
    Global Const KEY_NUMPAD6 = &H66
    Global Const KEY_NUMPAD7 = &H67
    Global Const KEY_NUMPAD8 = &H68
    Global Const KEY_NUMPAD9 = &H69
15  Global Const KEY_MULTIPLY = &H6A
    Global Const KEY_ADD = &H6B
    Global Const KEY_SEPARATOR = &H6C
    Global Const KEY_SUBTRACT = &H6D
    Global Const KEY_DECIMAL = &H6E
    Global Const KEY_DIVIDE = &H6F
20  Global Const KEY_F1 = &H70
    Global Const KEY_F2 = &H71
    Global Const KEY_F3 = &H72
    Global Const KEY_F4 = &H73
    Global Const KEY_F5 = &H74
25  Global Const KEY_F6 = &H75
    Global Const KEY_F7 = &H76
    Global Const KEY_F8 = &H77
    Global Const KEY_F9 = &H78
    Global Const KEY_F10 = &H79
    Global Const KEY_F11 = &H7A
30  Global Const KEY_F12 = &H7B
    Global Const KEY_F13 = &H7C
    Global Const KEY_F14 = &H7D
    Global Const KEY_F15 = &H7E
    Global Const KEY_F16 = &H7F
35  Global Const KEY_NUMLOCK = &H90

Function CategoryString (typeCode As Integer, catCode As Integer) As String
'creates user-readable string for a TV program's category
40  Dim msg As String

    msg = "Category: "
    'look up type code
    typeTable.Index = "ID"
    typeTable.Seek "=", typeCode
45  If typeTable.NoMatch Then
        msg = msg & typeCode
    Else
        msg = msg & typeTable("Name")
    End If
    msg = msg & ". " 'all on one line, replaced: Chr(13) & "Subcategory: "
50

```

55

```

        DayString = "Tuesday"
    Else
        DayString = "Tue"
    End If
5    Case 4
        If length = "long" Then
            DayString = "Wednesday"
        Else
            DayString = "Wed"
10   End If
    Case 5
        If length = "long" Then
            DayString = "Thursday"
        Else
            DayString = "Thur"
15   End If
    Case 6
        If length = "long" Then
            DayString = "Friday"
        Else
            DayString = "Fri"
20   End If
    Case 7
        If length = "long" Then
            DayString = "Saturday"
        Else
            DayString = "Sat"
25   End If
    End Select
End Function

30  Function fixAmpersand (text As String)
    'put in a "&" for every "&" so ampersand will print instead of format an underline
    Dim i As Integer
    Dim oldText As String
    Dim newText As String
35
    newText = ""
    oldText = text
    While InStr(cldText, "&")
        i = InStr(oldText, "&")
        newText = Left(oldText, i - 1) & "&&"
40
        oldText = Right(oldText, Len(cldText) - i)
    Wend
    fixAmpersand = newText & oldText
End Function

45  Sub InvokeHelp ()
    'add parameter for current location or give each form a local InvokeHelp
    'would be specialized for each view, probably not each button
    TellUser "Press Help (?) again for general help, or press any button on the
    remote for help with that button."
        Select Case returnCode
50    Case B_HELP

```

55

```

viewFilter = "Year >= 1993"
currView(MOVIE) = "Recent Movies"
currFilter(MOVIE) = ": All Categories"
5 SetStatus "Movies", greyCOLOR
Load listFrm(MOVIE)
'Shopping forms
currDomain = SHOP
filters(SHOP) = ""
SetStatus "Shopping, compact disks", greyCOLOR
10 Load listFrm(SHOP)
'TV forms
currFilter(TV) = "Basketball"
currDomain = TV
filters(TV) = "Category = 39"
15 userString = "Nova"
Load frmWeek
Load listFrm(TV)
Load frmComing
Load frmWkday
Load frmSelect
20
'show main menu
SetStatus "Use arrows and select or use keypad.", greyCOLOR
frmDex.Show
End Sub
25 Function Overlap (beginTS, endTS) As String
'create query string to look for TV programs in the range between
' and including beginTS and endTS
' Overlap = "{StartTS <= " & Str(endTS) & " And FinishTS >= " & Str(beginTS) & "}"
End Function
30
35 Sub SetInfo (text As String, Color)
'update the info box text and color
Dim s As SSSPanel
Set s = frmFrame!sspInfo 'works as long as form is loaded
s.BackColor = Color
s.Caption = text
End Sub
40
45 Sub SetKeys (mode As String)
'Set the keymappings for keyboard or "remote"
B_1 = KEY_NUMPAD7
B_2 = KEY_NUMPAD8
B_3 = KEY_NUMPAD9
B_4 = KEY_NUMPAD4
B_5 = KEY_NUMPAD5
B_6 = KEY_NUMPAD6
B_7 = KEY_NUMPAD1
B_8 = KEY_NUMPAD2
B_9 = KEY_NUMPAD3
If mode = "TV" Then
    'use keypad for all buttons (except 1-9)
    B_BACK = KEY_SUBTRACT
50
55

```

```

    statTable.Seek "=", s
    If statTable.NoMatch Then
        MsgBox "illegal station ID " & s
        Stop
    End If
    StationString = statTable("Name")
End Function

10 Sub TellUser (message As String)
    'displays message on screen until key is pressed
    'probably would not be used
    userMsg = message
    Wait frmMsg
End Sub

15 Function TimeLabel (t) As String
    'returns null string for times on half hour,
    ' returns hour 1..12 otherwise
    Dim s As String
    s = Format(t, "hh:mm AM/PM")
    If Mid(s, 4, 2) = "30" Then
        TimeLabel = ""
    Else
        s = Format(s, "h AM/PM")
        'strip off AM/PM
    25 TimeLabel = Left(s, Len(s) - 3)
    End If
End Function

Function TimeString (aDate) As String
    'format a date as 12-hour time without AM/PM or leading zero
    Dim theTime As String
    theTime = Format(aDate, "hh:mm AM/PM")
    theTime = Left(theTime, 5) 'take just "hh:mm" part
    If Left(theTime, 1) = "0" Then
        theTime = Right(theTime, 4)
    End If
    TimeString = theTime
End Function

Sub Wait (F As Form)
    'Allows one form to wait for another to hide itself
    F.Show
    While (F.Visible)
        DoEvents
    Wend
End Sub

45 '===== WEEK form code =====
Option Explicit
'stacked channel' view to be used with TV search and
' possibly other minimal searches (would need modification in ApplyFilter)
50 Dim allData(9) As snapshot  'all data within time period

```

55

```

'move to later time, same day
F.FindNext "StartTS > " & Str(TS)
success = Not F.NoMatch
If success Then
  'check if info arrows needed
  TS = F("StartTS")
  F.MoveNext
  If Not F.EOF Then
    If F("StartTS") = TS Then
      infoArrows "down"
    Else
      infoArrows "none"
    End If
  Else
    infoArrows "none"
  End If
  F.MovePrevious
End If
Case "Left"
'move to earlier time, same day
F.FindPrevious "StartTS < " & Str(TS)
success = Not F.NoMatch
If success Then
  TS = F("StartTS")
  'go to top of column
  F.FindFirst "StartTS = " & Str(TS)
  TS = F("StartTS")
  'check if info arrows needed
  F.MoveNext
  If Not F.EOF Then
    If F("StartTS") = TS Then
      infoArrows "down"
    Else
      infoArrows "none"
    End If
  Else
    infoArrows "none"
  End If
  F.MovePrevious
End If
Case "Down"
'move to later day, trying to keep close to previous time slot
If NProgs < 1 Then Exit Sub 'do nothing if all snapshots empty
aDay = aDay + 1: TS = TS + 48
While Not success And aDay <= NDays
  Set F = filterData(aDay)
  F.FindFirst "StartTS > " & Str(TS)
  If F.NoMatch Then
    'no prog to right, look left for any programs
    If Not F.EOF Then F.MoveLast
    If Not F.EOF Then
      success = True
      TS = F("StartTS")
    End If
  End If

```

55

```

      If Not F.EOF Then
          success = True
          TS = F("StartTS")
      End If
      5 Else
          'save program to right, count time slots away, look left
          marker = F.Bookmark
          best = F("StartTS") - TS
          F.FindLast "StartTS <= " & Str(TS)
          If F.NoMatch Then
              'no prog to left, take program to right
              F.Bookmark = marker
              TS = TS + best
          Else
              'check distances
              10 If TS - F("StartTS") > best Then
                  'right prog closest
                  F.Bookmark = marker
                  TS = TS - best
              Else
                  'left prog closest
                  TS = F("StartTS")
              End If
          End If
          'either way, we found a program
          success = True
          15 End If
          aDay = aDay - 1: TS = TS - 48
      Wend
      aDay = aDay + 1: TS = TS + 48
      If success Then
          20 'make sure to be at the top of a column
          F.FindFirst "StartTS = " & Str(TS)
          If F.NoMatch Then Stop 'how did we get a TS with no program in it?
          TS = F("StartTS")
          'check if info arrows needed
          25 F.MoveNext
          If Not F.EOF Then
              If F("StartTS") = TS Then
                  infoArrows "down"
              Else
                  infoArrows "none"
              End If
          End If
          30 Else
              infoArrows "none"
          End If
          F.MovePrevious
      End If
      35 Case "Next"
          'find next program, same time and day
          F.MoveNext
          If Not F.EOF Then
              40 'success means still in same time slot
              success = F("StartTS") = TS
          End If
      End If
      45
      50

```

55

```

Sub DisplayProg ()
  'set info box with current program info and highlight position
  Dim F As snapshot
  Dim msg As String

  Set F = filterData(currDay)
  msg = StationString(F("Station")) & ":" & Format(F("Start"), "h:mm AM/PM")
  msg = msg & " to " & Format(F("Finish"), "h:mm AM/PM. ")
  msg = msg & Format(F("Title"))
  msg = msg & Chr(13) & "(episode info here)" & Format(F("Episode"))
  'note: current database does not contain episode information
  SetInfo msg, Color(F(colorField) Mod 9)

  shpProg(0).Visible = False
  selector.Visible = False
  Position shpProg(0), F("StartTS"), F("FinishTS")
  CPlace 0, selector, shpProg(0)
  shpProg(0).Visible = True
  selector.Visible = True
End Sub

Sub DoPreview ()
  'Construct an appropriate preview message and display
  Dim msg As String

  msg = filterData(currDay)("Title")
  msg = msg & Chr(13) & "on " & StationString(filterData(currDay)("Station")) &
  Chr(13)
  msg = msg & CategoryString((filterData(currDay)("Type")),
  (filterData(currDay)("Category")))
  msg = msg & Chr(13) & DayString(Weekday(filterData(currDay)("Start")), "long")
  msg = msg & ", " & Format(filterData(currDay)("Start"), "mmm d, yy h:mm AM/PM")
  msg = msg & Chr(13) & " to " & Format(filterData(currDay)("Finish"), "h:mm
  AM/PM")

  popup.Caption = msg
  SizeAControl popup, (lblTime(1).top + 1.5 * lblTime(1).Height), 12,
  (lblDay(1).Width), 45
  popup.Visible = True
  inPreview = True
End Sub

Sub DoSelect ()
  'set data for selection and go to TV
  userStation = filterData(currDay)("Station")
  userStart = filterData(currDay)("Start")
  returnCode = TOTV
  Me.Hide
End Sub

Sub DrawProg (duplicates As Integer, index As Integer)
  'draw a program shape in display, marking it if there are duplicates at the
  identical time slot
End Sub

```

55

```

        ApplyFilter
        MakeDisplay
        sameFilter = True
5      End If
End Sub

Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    Dim index As Integer
    Dim n As Integer
10     Select Case KeyCode
        Case Asc("Q")
            End
        Case B_BACK
            returnCode = BACK
            Me.Hide
15        Case B_HELP
            sameFilter = True
            InvokeHelp
        Case B_PREVIEW
            If inPreview Then
                inPreview = False
                popup.Visible = False
            Else
                inPreview = True
            End If
20        Case B_RIGHT
            ChangeSel ("Right")
        Case B_LEFT
            ChangeSel ("Left")
        Case B_UP
            ChangeSel ("Up")
30        Case B_DOWN
            ChangeSel ("Down")
        Case B_SELECT
            If Not filterData(currDay).EOF Then DoSelect
        Case B_PAGEDOWN
            ChangeSel ("Next")
35        Case B_PAGEUP
            ChangeSel ("Prior")
        Case B_FILTER
            'go back to frmSelect to choose a new title
            returnCode = PICK
            Me.Hide
40        Case B_0
            returnCode = SHORTCUT
            Me.Hide
        End Select
45        If inPreview Then
            DoPreview
        Else
            popup.Visible = False
        End If
End Sub
50

```

EP 0 735 749 A2

```

d = 1
currDay = 1
While d <= NDays
    If filterData(d).EOF Then
        d = d + 1
    Else
        NProgs = 1 'just to make sure it is more than 0
        currDay = d
        d = NDays + 1
    End If
Wend
shpProg(0).ZOrder
selector.ZOrder
If Not filterData(currDay).EOF Then
    TScurrent = filterData(currDay).("StartTS")
    DisplayProg
    ChangeSel "none"
End If
End Sub

20 Sub Position (shape As Control, start, finish)
    'position a program shape
    Dim leftTS
    Dim rightTS
    Const smallGAP = .1

25    'convert to time slot scale
    leftTS = start - 48 * (currDay - 1)
    rightTS = finish - 48 * (currDay - 1)
    'set left and width
    shape.Left = dayWidth + leftTS
    shape.Width = rightTS - leftTS + 1 - smallGAP
30    'cut off at beginning of day
    'If shape.Left < dayWidth Then
        'shape.Width = shape.Width - (dayWidth - shape.Left)
        'shape.Left = dayWidth
    End If
35    'set top and height
    shape.Height = 2 - 2 * smallGAP
    shape.top = lblDay(currDay).top + smallGAP
End Sub

40 Sub SetInfo (msg As String, Color)
    'override the global SetInfo to write to my own info panel
    infoPanel.BackColor = Color
    infoPanel.Caption = msg
End Sub

45 '***** WKDAY form code *****
Option Explicit
'schedule of 5 weekdays at a particular time
'uses time-slot guided navigation

50 Dim allData(8) As snapshot 'all data within time period

```

55

```

'place program shapes
hasProgs = 0
c = 0 'init count of shpProgs
On Error GoTo ErrorHandler 'if we run out of shpProgs to allocate
5
For d = 1 To NDays
    currDay = d
    Set F = filterData(d)
    'create a shape control for each TV program in the data
    If Not F.EOF Then
10
        F.MoveFirst
        Do While Not F.EOF
            Load shpProg(c + 1)
            c = c + 1 'increment only after allocate succeeds
            shpProg(c).BackColor = Colcr(F(colorField) Mod 9)
            Position shpProg(c), F("Start"), F("Finish"), F("Station")
15
            shpProg(c).ZOrder
            shpProg(c).Visible = True
            F.MoveNext
        Loop
        F.MoveFirst
20
        If hasProgs = 0 Then hasProgs = d 'remember the first day with programs
        in it
        End If
        Next d
    MoveOn:
25
    On Error GoTo G 'quit trapping errors internally
    'make day lines visible on top
    For d = 1 To NDays - 1
        dayLine(d).ZOrder
        dayLine(d).Visible = True
    Next d
30
    'initialize stuff
    NProgs = c
    currDay = hasProgs
    shpProg(0).ZOrder
    selector.ZOrder
35
    If currDay > 0 Then
        'set time slot begin and end numbers for current day
        TSBegin = DateDiff("n", refDate, startTime) \ 30 + 48 * (currDay - 1)
        TSEnd = TSBegin + slotsPerDay - 1
        TScurrent = TSBegin
40
        Set F = filterData(currDay)
        Do While TScurrent <= TSEnd
            F.FindFirst Overlap(TScurrent, TScurrent)
            If Not F.NoMatch Then
                DisplayProg
                Exit Do
45
            End If
            TScurrent = TScurrent + 1
        Loop
    Else
50
        TSBegin = DateDiff("n", refDate, startTime) \ 30
        TSEnd = TSBegin + slotsPerDay - 1

```

55

```

        TScurrent = TSBEGIN
        currDay = 1
    End If
5     shpSlot(TScurrent - TSBEGIN + 1 + (currDay - 1) * slotsPerDay).FillStyle =
        0'solid
        lblTime(TScurrent - TSBEGIN + 1 + (currDay - 1) * slotsPerDay).BackStyle =
        1'opaque
        Exit Sub
10
ErrorHandler:
    If Err = 342 Then
        'ran out of room to allocate program shapes, quit drawing
        Resume MoveOn
    Else
15     Dim msg
        msg = Error & Chr(13) & "Resume or Cancel?"
        msg = InputBox(msg, "Error Correction", "Resume")
        If msg = "" Then Stop
        Resume MoveOn
    End If
20
End Sub

Sub Position (shape As Control, start, finish, station)
    'position a program shape for display
    Dim relativeL, relativeW, dayStart
25
    Dim edge
    'convert a day/time to position in NSlot scale
    dayStart = startTime + currDay - 1
    relativeL = (start - dayStart) * 48
    relativeW = (finish - dayStart) * 48 - relativeL
30
    'clip shapes off at day boundaries
    If relativeL < 0 Then
        relativeW = relativeW + relativeL
        relativeL = 0
    End If
    If relativeW + relativeL > slotsPerDay Then relativeW = slotsPerDay - relativeL
35
    'set left and width of shape
    edge = (currDay - 1) * slotsPerDay
    shape.Left = relativeL - edge + sideGap
    shape.Width = relativeW - 2 * sideGap
    'enforce minimum width so program is visible
40
    If shape.Width < MINProgWidth Then shape.Width = MINProgWidth
    'set top according to station
    'note: this scheme only works because stations are named 1..n
    rowOffset = ((500 - 2 * lblHEIGHT - shpProg(0).Height) / NSTATION)
    shape.Top = shpSlot(0).Top + topGAP + (station - 1) * rowOffset
    End Sub
45

```

50

Thus, it will now be understood that there has been disclosed a method and apparatus of finding and selecting a program to view from a large schedule of TV programs. While the invention has been particularly illustrated and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form, details, and applications may be made therein. For example, color coding of the individual items of the reduced representations and of the various entries in the various grid displays could be used to assist the viewer in making rapid program selections. Another example is that it is easily within the capabilities of this art to modify a TV set by integrating the set top box according to the present invention into it. It is accordingly intended that the appended

claims shall cover all such changes in form, details and applications which do not depart from the true spirit and scope of the invention.

5 **Claims**

10 1. Apparatus for selecting an item from a group thereof in a system having display means and interactive movable pointing means for specifying a location in the display means and making a selection at a specified location, the apparatus comprising:

15 filtration means including subgroup specifiers in the display means and responsive to selection of a subgroup specifier by the pointing means for filtering the group to produce the subgroup specified by the selected subgroup specifier;
means for displaying representations of group items belonging to at least a portion of the subgroup in the display means; and
group item selection means for selecting a group item by selecting the representation thereof in the display in response to the pointing means.

20 2. The apparatus set forth in claim 1 wherein:
the pointing means need only be movable from one representation to an adjacent representation.

25 3. The apparatus set forth in claim 1 wherein:
the means for displaying the representations comprises:
first means for displaying the representations in a single dimension; and
second means for displaying the representations in two dimensions.

30 4. The apparatus set forth in claim 1 further comprising:
means for displaying a reduced representation of the entire subgroup and an indication in the reduced representation of the portion of the group being presently displayed by the display means.

35 5. The apparatus set forth in claim 4 wherein said reduced representation is two dimensional.
6. The apparatus set forth in claim 5, wherein said interactive movable pointing means includes a remote control having:

40 a first pair of buttons to control changes in location in the display in a first direction; and
a second pair of buttons to control changes in location in the display in a second direction.
7. The apparatus set forth in claim 4 wherein said reduced representation is a two dimensional representation of a three dimensional representation, the third dimension being location within a logical stack of items having at least one common property.

45 8. The apparatus set forth in claim 7 wherein each item of a logical stack have viewing timeslot as one common property.

50 9. A method comprising the steps of:
receiving program schedule data by a set top box via a same information conductor that conducts program information to the set top box;
filtering said program schedule data in RAM within said set top box;
said set top box showing a first interactive display on a TV connected thereto presenting a plurality of choices for filtering said program schedule data to a viewer;
in response to an interactive selection by said viewer, filtering said program schedule data into a first subgroup of program schedule data;
also in response to an interactive selection by said viewer, said set top box showing a second interactive display on said TV having a second plurality of choices for filtering said program schedule data;
in response to a second interactive selection by said viewer, filtering said first subgroup into a second subgroup;

and

also in response to a second interactive selection by said viewer, said set top box showing a third interactive display on said TV having a representation of each program item of said second subgroup.

- 5 10. The method of claim 9, wherein said receiving program schedule data step further comprises the steps of:
 - receiving a first portion of said program schedule data via said set top box; and
 - receiving a second portion of said program schedule data at a later non-contiguous time.
- 10 11. The method of claim 9, further comprising the step of:
 - in response to an interactive highlighting of a representation of a program item of said second subgroup, displaying a title thereof.
- 15 12. The method of claim 11, further comprising the step of:
 - in response to an actuation of a select button of a remote control, displaying a preview of said highlighted program.
- 20 13. The method of claim 12, further comprising the step of:
 - in response to a second actuation of said select button of said remote control, switching said set top box to display a TV program corresponding to said highlighted representation.
- 25 14. The method of claim 12, further comprising the step of:
 - in response to a second actuation of said select button of said remote control, storing a command to switch said set top box to display a TV program corresponding to said highlighted representation in when that TV program begins.
- 30 15. A method comprising the steps of:
 - receiving program schedule data for at least 300 individual channels for a time period of at least a week;
 - storing said program schedule data in local memory for rapid sorting and retrieval in a database format;
 - filtering the program schedule data in response to interactive user inputs into a subgroup of the program schedule data;
 - displaying the subgroup of the program schedule data for the user's review; and
 - interactively selecting a program from the subgroup of program schedule data for viewing on a TV screen.
- 35 16. A method for choosing a desired program from a large schedule of programs whose data is stored in a local memory, comprising the steps of:
 - displaying a vertically cascaded group of cards with each card representing a program of a particular time and channel;
 - displaying a selection window located around a subgroup of said group of cards;
 - displaying a two-dimensional grid adjacent to said vertically cascaded group of cards in which said subgroup of the programs represented by said vertically cascaded group of cards are shown in greater detail;
 - displaying a first active area within said selection window highlighting one of said subgroup of programs;
 - displaying a second active area within said two-dimensional grid, said second active area being located around and highlighting greater details of the program highlighted in said first active area;
 - moving said first active area in a vertical direction in response to vertical direction arrows to a viewer's input of a remote control; and
 - selecting a desired program by moving said active area to said desired program and actuating a select button until said set top box makes said selection.
- 40 17. The method according to claim 16, further comprising the steps of:
 - after said active area is moved one location outside of said selection window by inputs from said viewer, moving said selection window to a contiguous subgroup to which said active area has moved.
- 45 18. Apparatus for selecting an item from a group thereof in a system having display means and interactive movable pointing means for specifying a location in the display means and making a selection at a specified location, the apparatus comprising:

EP 0 735 749 A2

filtration means including subgroup specifiers in the display means and responsive to selection of a subgroup specifier by the pointing means for filtering the group to produce the subgroup specified by the selected subgroup specifier;

means for displaying representations of group items belonging to the subgroup in the display means; group item selection means for selecting a group item by selecting the representation thereof in the display in response to the pointing means; and

means for displaying a reduced representation of the entire subgroup and an indication in the reduced representation of the portion of the group being presently displayed by the display means;

said reduced representation displaying means displaying a two dimensional representation of a three dimensional representation, a third dimension being represented as a logical stack of items having at least one common property.

19. The apparatus set forth in claim 8 or 18, wherein said interactive movable pointing means includes a remote control having:

15 a first pair of buttons to control changes in location in the display in a first direction; and
a second pair of buttons to control changes in location in the display in a second direction; and
a third pair of buttons to control changes in location within the logical stack.

20

25

30

35

40

45

50

55

FIG. 1

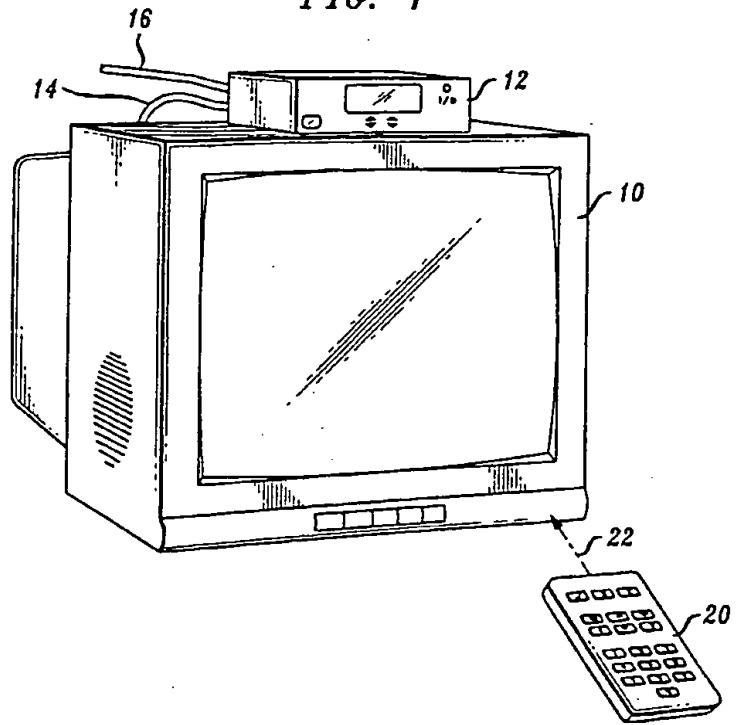


FIG. 2

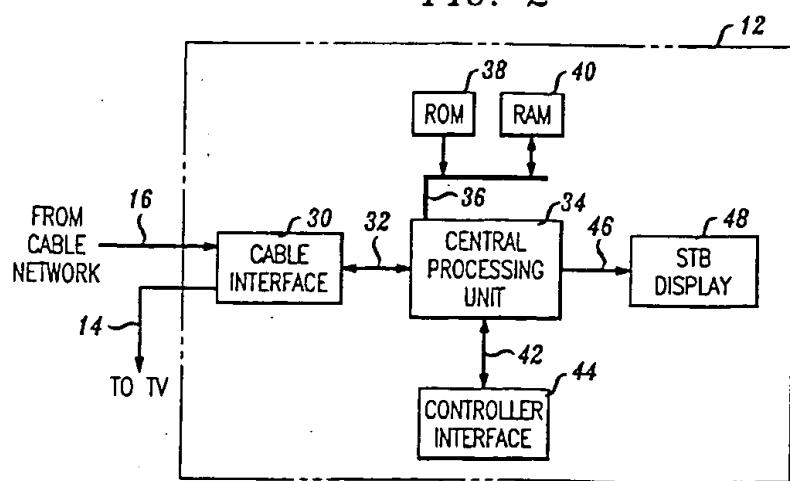


FIG. 3

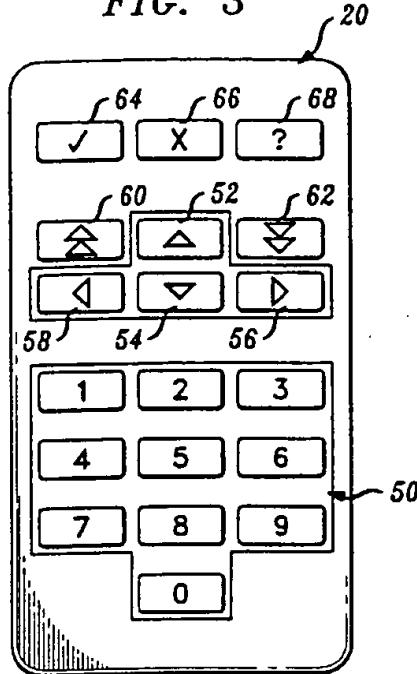


FIG. 4

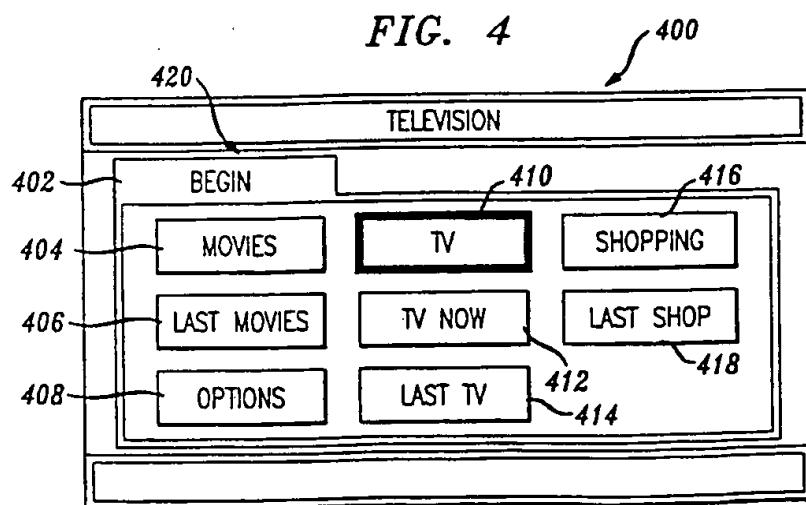


FIG. 5

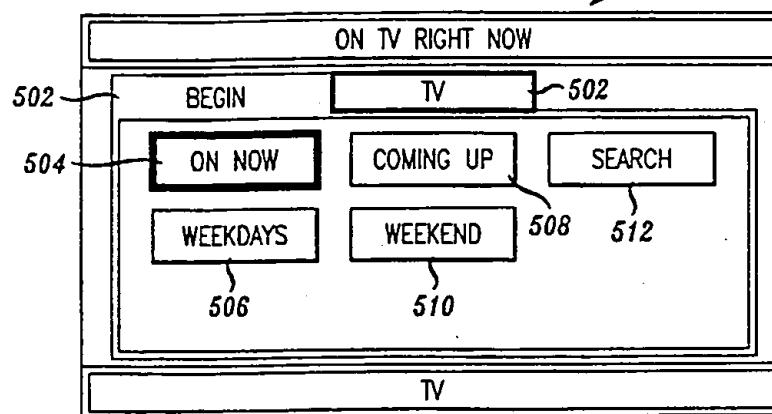


FIG. 6

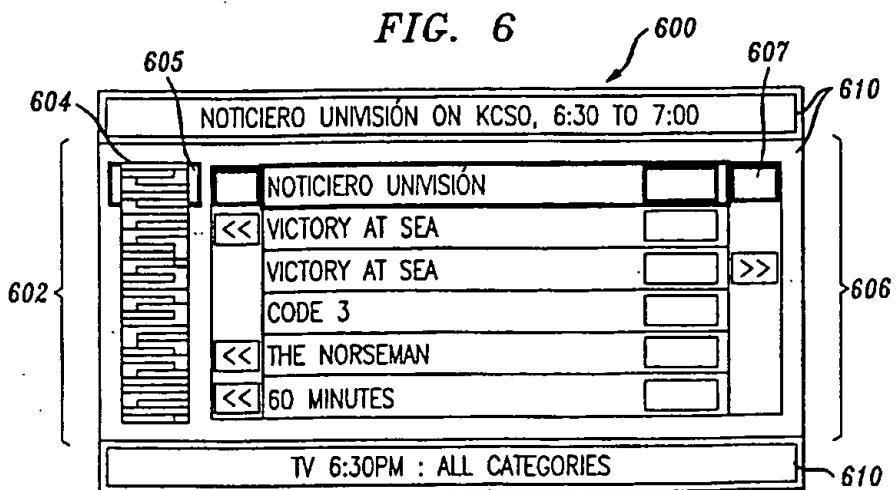


FIG. 7

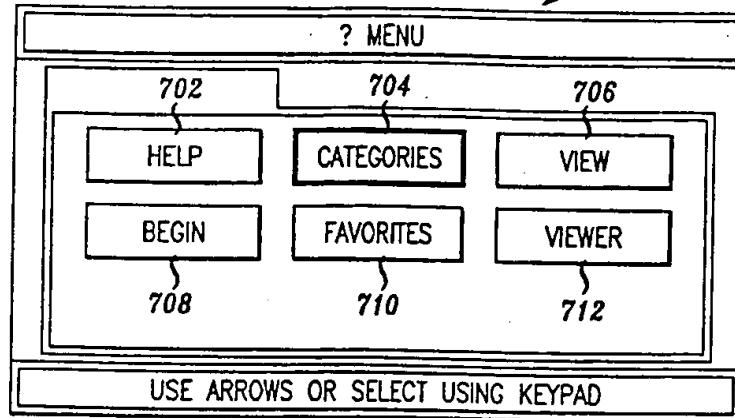


FIG. 8

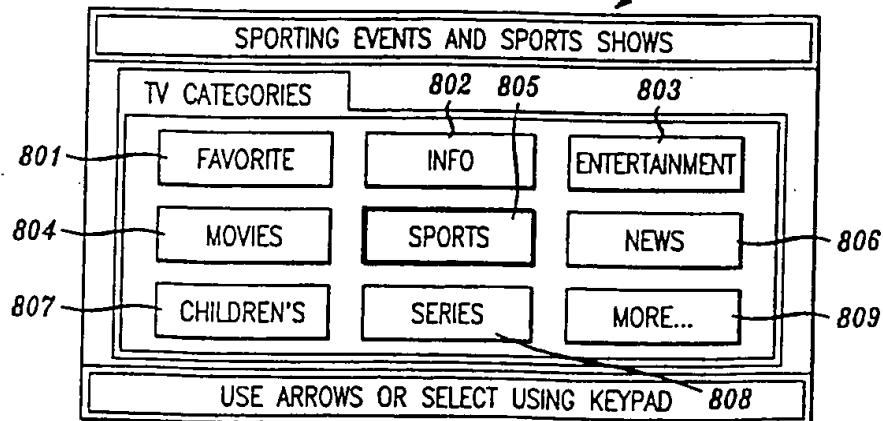


FIG. 9

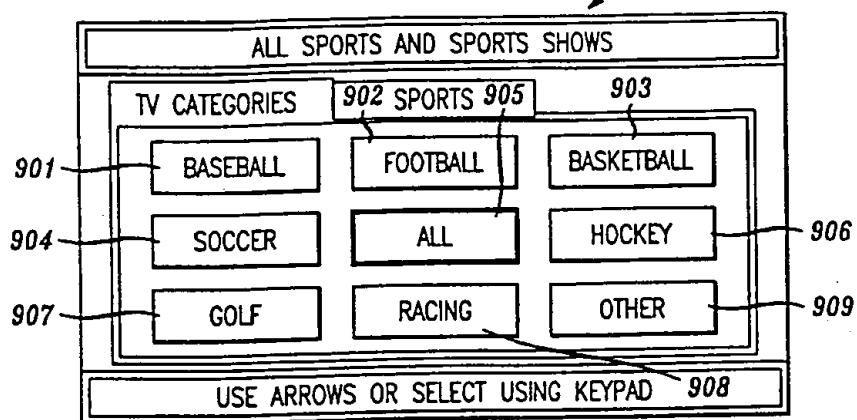


FIG. 10

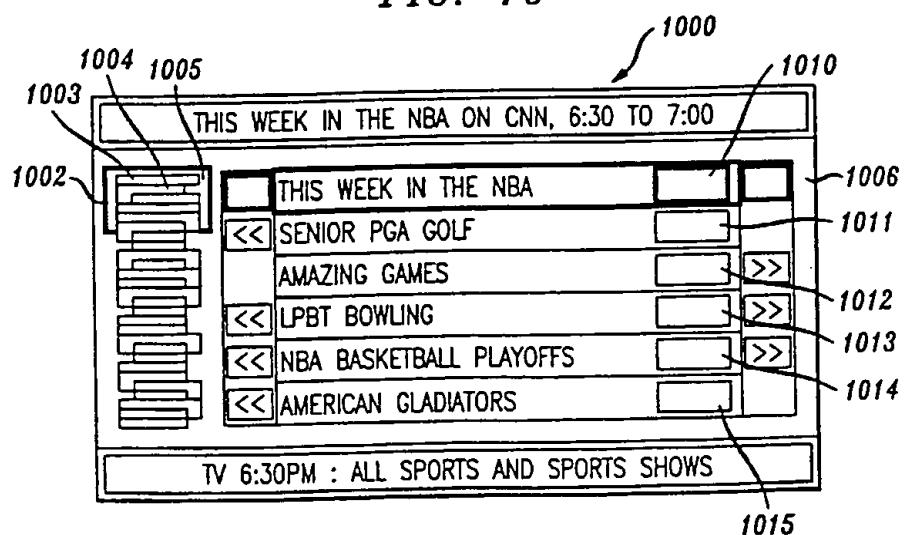


FIG. 11

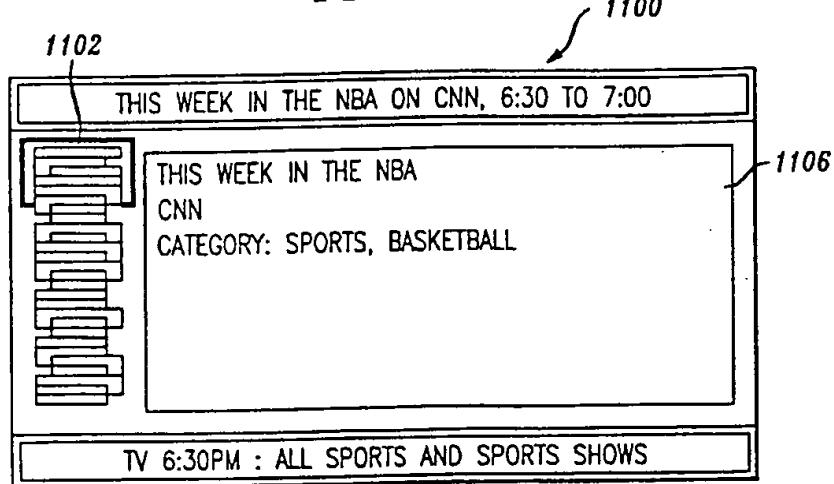


FIG. 12

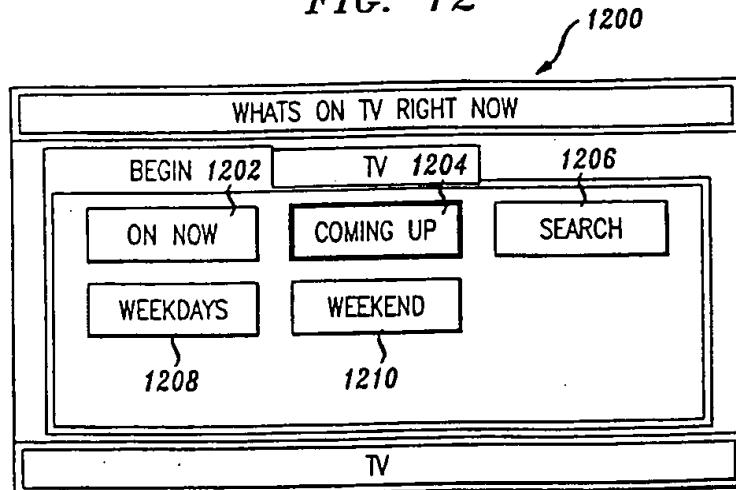


FIG. 13

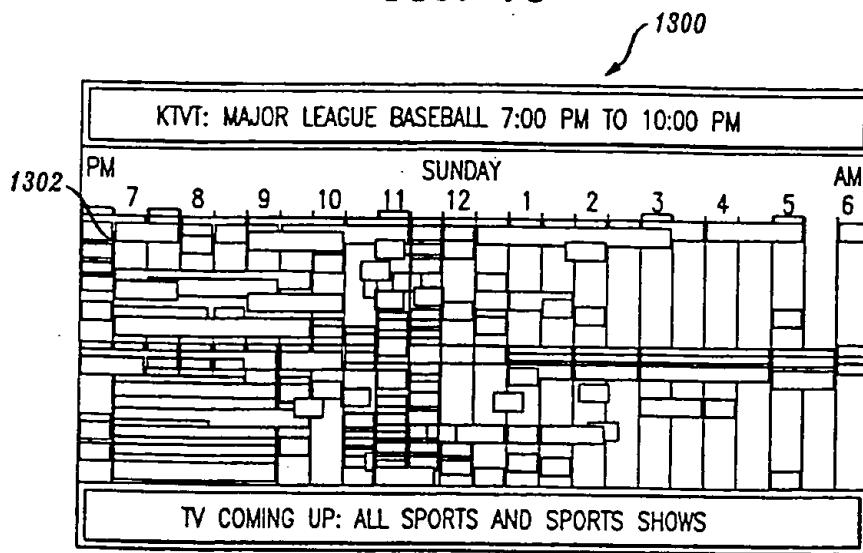


FIG. 14

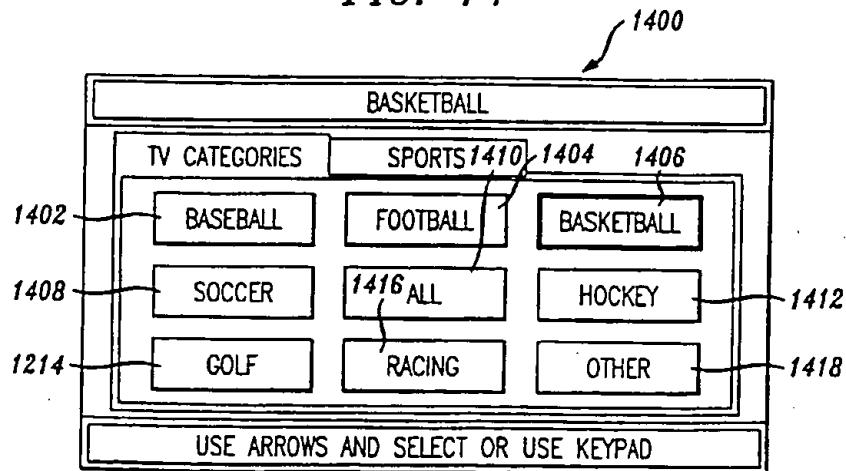


FIG. 15

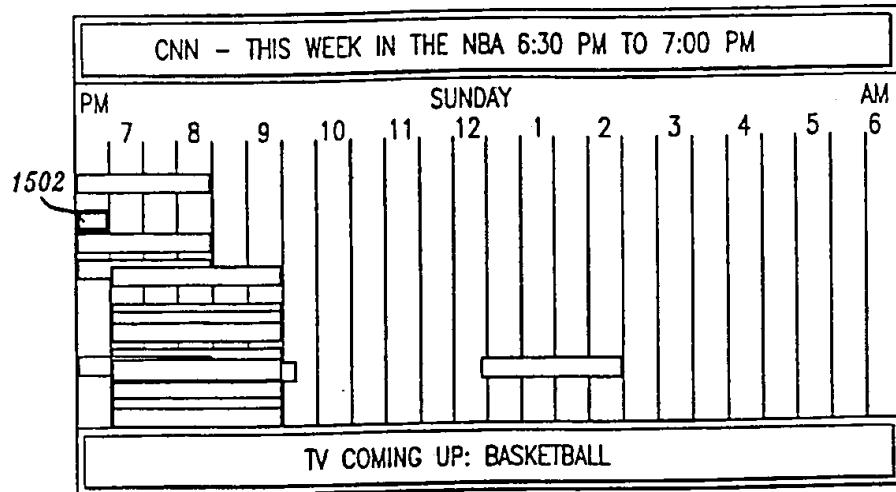


FIG. 16

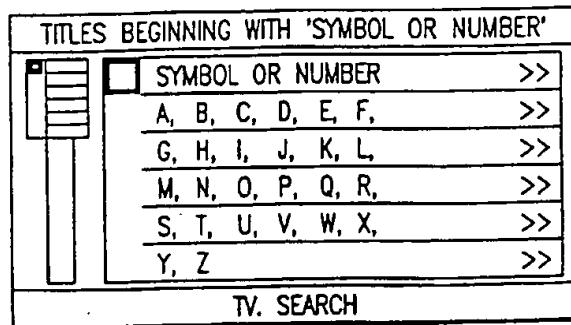


FIG. 17

1700

TITLES BEGINNING WITH 'M, N, O, P, Q, R'	
<input checked="" type="checkbox"/>	SYMBOL OR NUMBER >>
<input type="checkbox"/>	A, B, C, D, E, F, >>
<input type="checkbox"/>	G, H, I, J, K, L, >>
<input checked="" type="checkbox"/>	M, N, O, P, Q, R, >>
<input type="checkbox"/>	S, T, U, V, W, X, >>
<input type="checkbox"/>	Y, Z >>
TV. SEARCH	

FIG. 18

1800

TITLES BEGINNING WITH 'M'	
<input checked="" type="checkbox"/>	M >>
<input type="checkbox"/>	N >>
<input type="checkbox"/>	O >>
<input type="checkbox"/>	P >>
<input type="checkbox"/>	Q >>
<input type="checkbox"/>	R >>
TV TITLES STARTING WITH M, N, O, P, Q, R	

FIG. 19

1900

TITLES BEGINNING WITH 'N'	
<input checked="" type="checkbox"/>	M >>
<input checked="" type="checkbox"/>	N >>
<input type="checkbox"/>	O >>
<input type="checkbox"/>	P >>
<input type="checkbox"/>	Q >>
<input type="checkbox"/>	R >>
TV TITLES STARTING WITH M, N, O, P, Q, R	

FIG. 20

2000

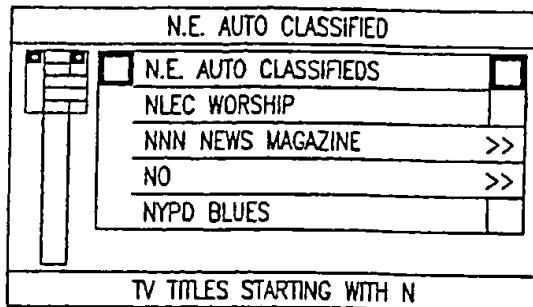


FIG. 21

2100

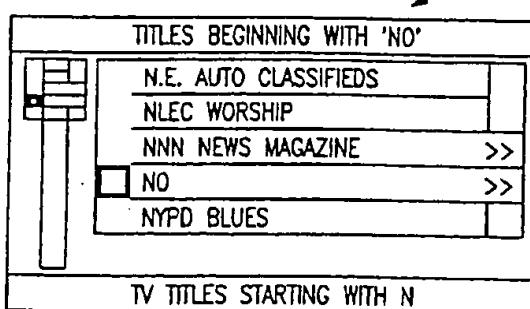


FIG. 22

2200

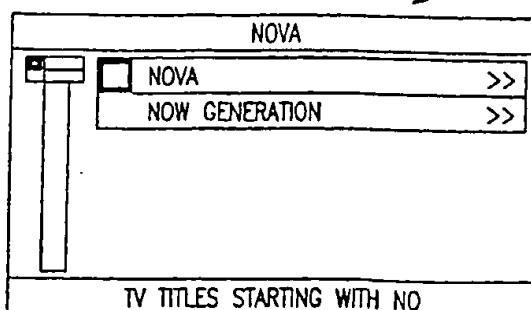


FIG. 23

WLVT: 12:00 AM TO 1:00 AM, NOVA (EPISODE TITLE)												
2302	AM				NOON				PM			
	12	1	2	3	4	5	6	7	8	9	10	11
SUN									*			*
MON				*								*
TUE										*	*	
WED									*			*
THUR								*				*
FRI												
SAT										*		

THIS WEEK: NOVA